Design • Build • Program • Learn • Compete

SERVO

FOR THE ROBOT INNOVATOR
www.servomagazine.com

MAGAZINE
November 2012

Rock And Rover

Adding An Arm Mechanism To The MINDS-i LUNAR ROVER

- Build A Quadcopter
- Telebot - Arduino Controlled Bot Via Zigbee

Reviews Of The

- RoboVoice Chip
- ArduPilot 2.5

U.S. $5.50 CANADA $7.00

11

0 71486 02422 4

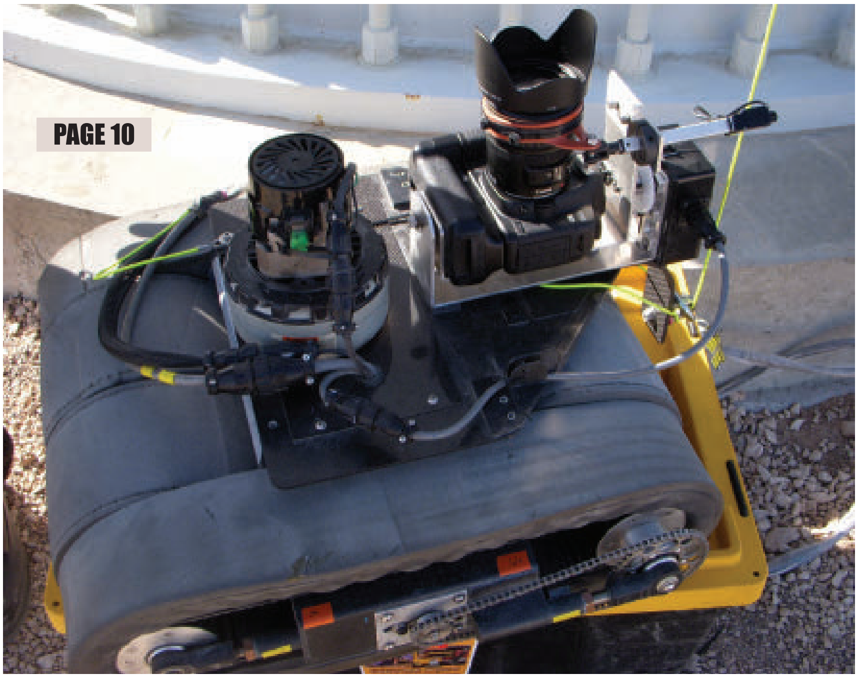

X1 AC PLUS X2 ULTIMA X2 400 Four Hundred X4 AC PLUS X4 80 Eighty
NOW IS NOT THE TIME TO WORRY ABOUT YOUR BATTERY!
No matter what type of batteries you use, Hitec has a charger for you! Choose from the portable X1 AC Plus, the efficient X2 Ultima, the four-channel X4 AC Plus, the revved up X4 Eighty or the powerful X2 400. That's right, our line offers five dependable battery chargers to get you charged up and ready for battle!
HITEC
12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrcd.com
Photo courtesy of Sam Coniglio
X1 AC PLUS X2 ULTIMA X2 400 Four Hundred X4 AC PLUS X4 80 Eighty

Simple High-Power Motor Controller 18v25 and 24v23 Pin-Out
USB Interface (PC)
RC Interface (RC Receiver)
TTL Serial Interface (Microcontroller)
Analog Interface (Joystick/Potentiometer/Limit Switch)
Battery
SIMPLE MOTOR CONTROLLER?
SERVO CONTROLLER?
USE COUPON CODE! SERVO8693
WIRELESS ROUTER
SONAR RANGE FINDER??
DISTANCE SENSOR?
SERVO
MOTOR CONTROLLERS
VOLTAGE REGULATOR
Take your design from idea to reality
Pololu
Engage Your Brain
www.pololu.com

SERVO MAGAZINE

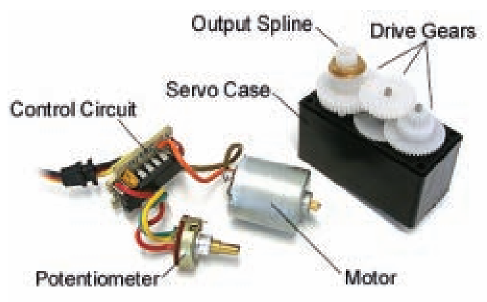
11.2012

VOL. 10 NO. 11



PAGE 10

Columns



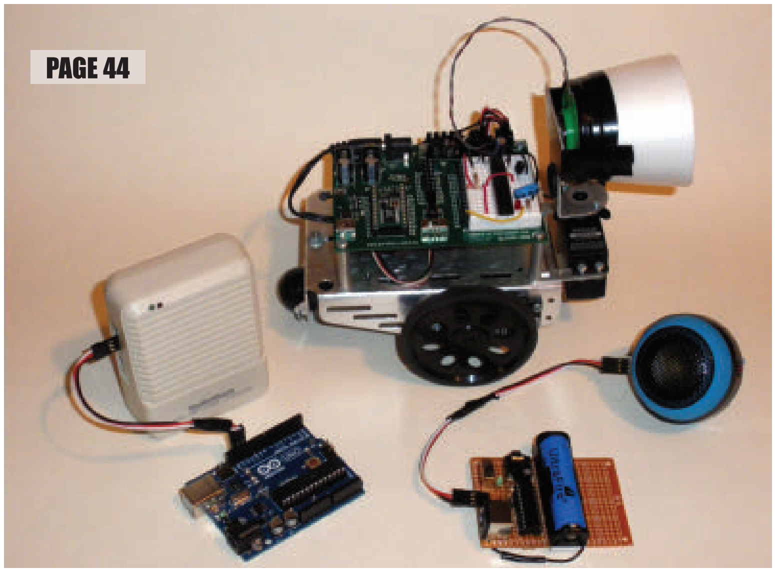
PAGE 74

Departments

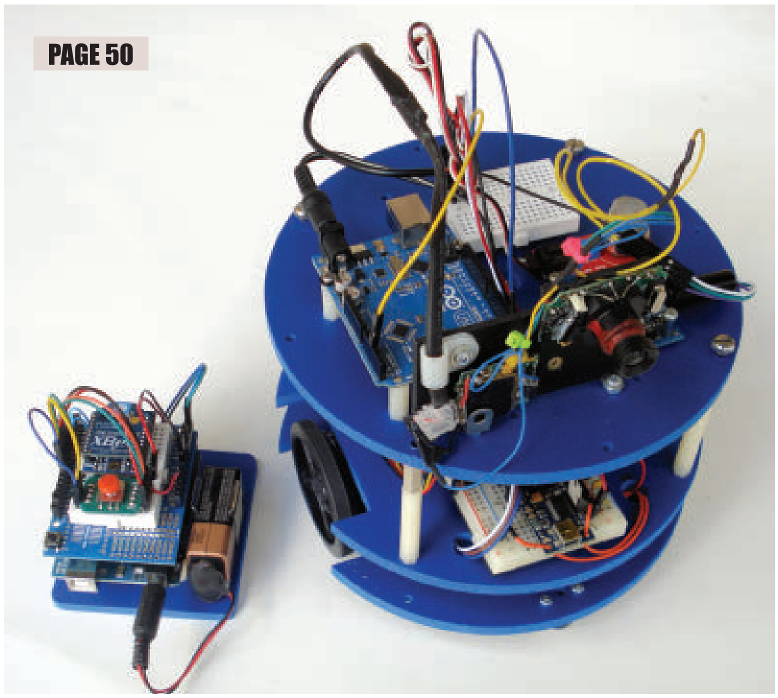
PAGE 24

SERVO Magazine **(ISSN 1546-0592/CDN Pub Agree#40702530)** is published monthly for $24.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL ENTRY MAILING OFFICES. POSTMASTER: Send address changes to **SERVO Magazine, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@servomagazine.com

In This Issue ...

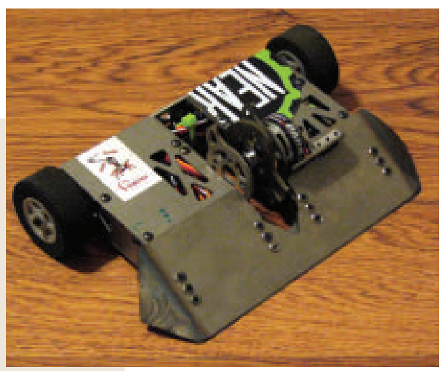


PAGE 44



PAGE 50

The Combat Zone...



by Bryan Bergeron, Editor

Curiosity

The successful landing of NASA's Curiosity on Mars is arguably the most significant robotics event of the decade. Not only have the stunning photographs at Bradbury Landing and the back story of the mission captured the imaginations of thousands of future scientists and engineers, but the value of robotics has been demonstrated to the public.

As a roboticist, there's a lot to learn from the project. The most significant lesson is probably that you don't have to wear a pocket protector to make a contribution to the field. NASA's PR team has been highlighting the career paths of several key players in the project who don't fit the expected stereotype roborocket scientist. For example, the leader of the Mars Lander component of the project is a would-be rock-and-roll star who barely passed high school. To satisfy his personal curiosity, he enrolled in a physics course at a community college, and his new trajectory was set.

Another lesson is the power of teams to tackle seemingly insurmountable obstacles. Although there were project leaders, the heavy lifting was performed by thousands of engineers and scientists over many years. Designing the Lander was a ten year project that at times involved 2,000 engineers, for example.

The mission also highlights the importance of simulation in the R&D process. The Lander was never subjected to a full physical test, but the design group relied on simulations and avoided a costly and lengthy build-crash-rebuild cycle.

Another take-away is start off by taking baby steps. After landing the six-wheeled vehicle on Mars, the scientists back at NASA didn't take it for a remote joy ride across the Gale Crater. Instead, they tested the sensors (at least one wind sensor was found to be inoperative) and then made a short, slow, 16 minute excursion. There's no room for error, and one wrong turn could result in a flipped vehicle and the end of the mission.

If I had to rank the lessons learned from the Curiosity project, I'd have to pick starting slowly and carefully as the most important take-away. If you're working with a $600 quadcopter and you accidently pilot it into a telephone pole at 40 mph, you'll have to restart from scratch. Better to start off with small, short hops until you get a feel for the craft and the controller. It's the same with a high-speed battle-bot — you can't rush testing when you're dealing with high-speed spinning discs, hammers, and other tools of destruction.

From a strictly mechatronic perspective, there's a wealth of information that you can gain by studying the images provided by NASA. For example, take a look at the propulsion system, including the wheels that can rotate freely about the vertical axis in a sort of 'C Clamp' cage. What have you borrowed from Curiosity's design to make your next robot, whether it's a simple terrestrial roamer, high-speed battle-bot, or a quadcopter a success? SV

FOR THE ROBOT INNOVATOR
SERVO MAGAZINE

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct., Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore Only **1-800-783-4624**
www.servomagazine.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277, N. Hollywood, CA 91615

PUBLISHER
Larry Lemieux
publisher@servomagazine.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@servomagazine.com

EDITOR
Bryan Bergeron
techedit-servo@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Jenn Eckert, Tom Carroll, David Geer, Dennis Clark, R. Steven Rainwater, Kevin Berry, Michael Jeffries, Dave Graham, Pete Smith, Gordon McComb, Eric Ostendorff, Michael Simpson, Bryce Woolley, Evan Woolley

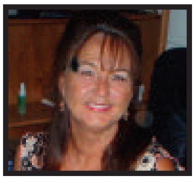
CIRCULATION DEPARTMENT
subscribe@servomagazine.com

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@servomagazine.com

WEB CONTENT
Michael Kaudze
website@servomagazine.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Sean Lemieux

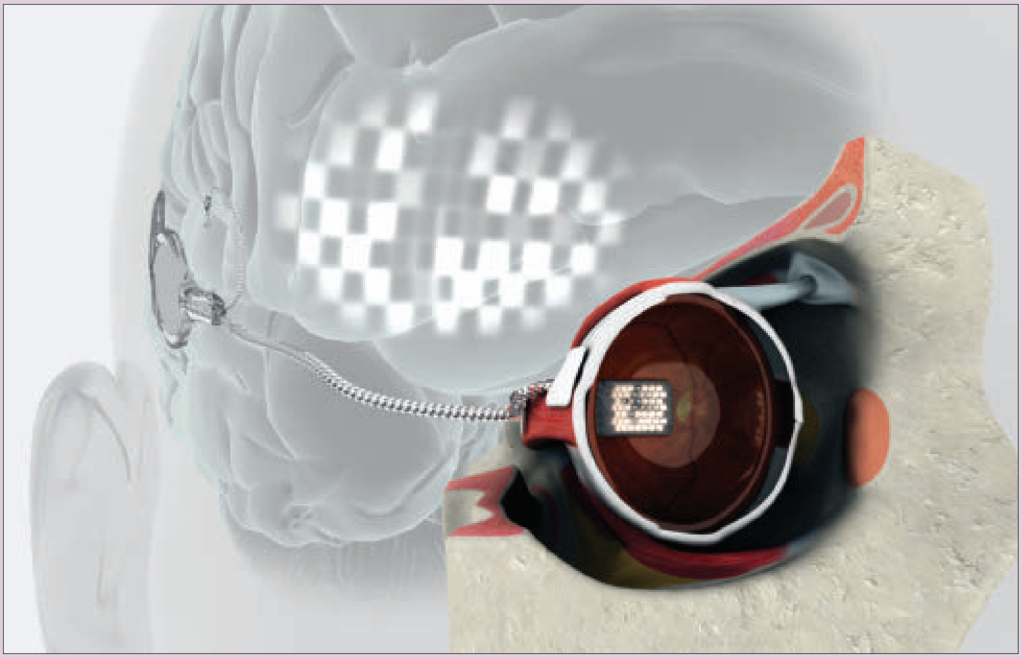
Robytes

by Jeff and Jenn Eckert

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com.

Now See Here

In 2009, the Australian government awarded a $42 million grant to Bionic Vision Australia (BVA, **www.bionicvision.org.au**) — a national consortium of researchers working to develop a practical bionic eye. The goal is to restore sight to people who suffer from such retinal degenerative conditions as retinitis pigmentosa and macular degeneration. The long-term solution involves a wireless implant that uses 1,024 stimulating electrodes made of a specially designed diamond material that is unaffected by moisture, and that has no inflammatory effects on the human body. In a major step toward that goal, BVA researchers recently performed the first implantation of an early prototype. The "pre-bionic eye" has only 24 electrodes and doesn't really produce anything that qualifies as vision, but patient Dianne Ashworth verified that it worked when switched on. "All of a sudden I could see a flash of light," she commented. "It was amazing."



Artist's rendering of the BVA bionic eye.

According to surgeon Dr. Penny Allen, "This is a world first — we implanted a device in this position behind the retina, demonstrating the viability of our approach." The final version is expected to produce clear enough images to allow recipients to recognize faces and read large print.

The QBotix Tracking System dynamically operates solar power plants to maximize output.

Solar Efficiency on Track

One might imagine that the most economical way to keep solar panels aimed toward the sun would be to fit them with standard single- or dual-axis tracking systems, but QBotix (**www.qbotix.com**) begs to differ. According to the company, the QBotix Tracking System™ offers the performance of dual-axis systems at a single-axis price by replacing the hundreds of motors and controllers found on conventional tracking systems. The system — designed to handle 300 kW installations (about 200 panels) — uses two battery-operated monorail bots (one active, one backup) which travel around and successively adjust each panel's mounting system. According to QBotix, the system can track both flat-plate and concentrating solar panels with high accuracy and reliability, resulting in a 20 percent lower "levelized cost of electricity." According to BEO Wasqu Bokhari, "Regardless of the choice of solar panels, inverters, foundations, or other system components, the use of QTS will dramatically lower LCOE compared to all existing mounting or tracking systems."

Snake Bites with Laser

OC Robotics' Explorer snakebot armed with a 5 kW laser, integrated navigation camera, and lighting.

If you have some work to do in a hazardous location, you might send in some autonomous bots to do the work. If the job requires some serious electrical power that battery-driven units can't provide, it may be a better idea to poke a snakebot through an available aperture. This is the idea behind the LaserSnake project put together by OC Robotics (**www.ocrobotics.com**) with collaborator TWI Ltd. (**www.twi.co.uk**) as part of a UK Technology Strategy Board competition for nuclear R&D feasibility studies. A snake-arm robot was combined with a 5 kW laser to create a selective, remote-controlled approach to dismantling and decommissioning complex structures in hazardous and confined nuclear environments. In a recent demonstration, one of OC's Explorer models demonstrated the ability to perform remote single-sided cutting using a fiber laser — a trick that could ultimately be applied in a real world nuclear environment to dismantle vessels, support structures, flasks, and pipe work. With the ability to cut through pressure vessels, I-beams, boxes, and tubes, this guy has a bite that even a Black Mamba would envy. To see one of the units doing repair work, just slither on over to **www.youtube.com/watch?v=KsNeLB-EHsg**.

Bots to Adorn Your Fingers

Sometimes you get the feeling that robot developers in Japan are intentionally putting us on, or maybe just having some fun coming up with silly things and inventing explanations later. The suspicion grows with a set of robotic rings developed by a "research group" at Keio University (**www.keio.ac.jp**). Each unit contains its own motor, but a PC-driven microcontroller determines movement of the eyes and mouth from afar.

Robotic rings developed at Keio University.

Of what practical use could this be, you ask? Well, according to the group, "When you wear this robot on your hand, it forms a medium for communication using the hand. So, the robot serves as a device for enhancing the animal-like, imitative ways that people use their hands." I can think of a way to communicate with them using my hand and no rings, but never mind. Just don't offer a set to your fiancee if you want the answer to be "yes."

Bot Knows Squat

Squat lobsters crawling around in a *Lophelia* mound. (Image courtesy of Deep Sea Systems, Schmidt Ocean Institute.)

Exploring a different kind of hostile environment is the work of the Schmidt Ocean Institute (**www.schmidtocean.org**), whose purpose in life is "to advance ocean exploration, discovery, and knowledge, and catalyze sharing of information about the oceans." At the end of August, the Schmidt folks dropped a Global Explorer MK3 ROV into the Okeanos Ridge site — a previously unexplored 2,000 ft (600 m) deep seafloor area located about 185 mi (300 km) west of the Florida gulf coast. The MK3 more or less resembles a rectangular frame with two tanks and a bunch of cameras mounted on it, so we're not bothering to show it here. However, it is one of the few ROVs that is able to record 3D video, as well as high-def 2D video, and it came up with many fascinating images of things like living corals, chirostylid and golden crabs, and (see photo) squat lobsters. A visit to the website's gallery is worth the trip. SV

GEEK HEAD

by David Geer

Contact the author at geercom@windstream.net

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

International Climbing Machines Robots Go Places Humans Can't

International Climbing Machines (ICM) robots use vacuum power to stay adhered to tall surfaces as they maneuver about, taking care of inspections and a variety of tasks. The robots can inspect other machines and surfaces using a video camera payload and when equipped with lasers, these climbers can clean, decontaminate, and prepare surfaces. Companies can attach paint sprayers, abraders, and other tools for a variety of applications.

What Exactly are ICM Robots?

ICM climbing robots are remote controlled from a laptop via a tether and can work, inspect, and clean where great heights, contaminants, toxins, fatigue, and other environmental risks could harm a human laborer.

The robots cling to any vertical or inverted surface including brick, concrete, and metal, using a unique patented vacuum force seal that holds the full weight of the robot — including payloads up to a total of 225 lbs. The climber uses two drive motors and one vacuum motor.

The seal keeps the machine attached to surfaces even as it climbs and crawls past plates, welds, or seams. The vacuum is also useful for actually cleaning the surfaces as it removes debris and then funnels it through a hose into drums. The relatively small, quickly deployable (30 minutes) light (30 lbs) climbing robots can haul large payloads, including interchangeable attachments such as Roto Peen or Starcutter coating removal tools, blast media coating removal tools, non-abrasive brushes, NDT and NDE inspection tools, and tools that measure coating thickness, wall thickness, corrosion, and stress-related surface cracks.

This close-up shows the adhesion vacuum track technology that enables the climbing robot to adhere to uneven and contoured surfaces, and to carry its own weight as well as payloads used for a variety of inspection applications.

Robot Construction

The ICM climbing robots are made of advanced composites and carbon fiber, and measure 24 inches by 24 inches by 8 inches tall. The abrader control assembly is 40 lbs with a 6-12 inch cleaning path

width. The climbers travel at about three inches per second, which is not bad for a robot that won't fall off of a given surface. The robot rides on durable foam tracks.

These climber bots have been used for both wind turbine blade and tower inspection and repair. These bots are also useful for inspecting bond lines between the surface of the windmill blade and the support beam inside the blade. Climbers have performed ultrasonic inspections of trailing edges of wind turbine blades.

This unique robot is also useful for hot cell remote spraying of mastic coatings using a spray boom. Climbers have been used in nuclear containment deployments. The ICM robots do not scratch or damage any surfaces.

ICM bots can inspect and repair in most kinds of weather. They replace the legacy process of waiting for several wind turbines to go down before using a crane to inspect and repair them. Additionally, the robots offer permanent digital mapping and recording of all inspections and jobs. Plus, these robot climbers do not require scaffolding and can perform tests quickly at a reasonable cost.

Nuclear Applications

Issues with coating failures challenge the nuclear industry; corrosion and degradation of concrete and metal structures and the condition of the structures cannot be verified through firsthand human means. Inspectors have to either use binoculars or high magnification cameras that record and map visual feedback.

Regulations require the nuclear power industry to verify containment integrity, which cannot be completed by cameras only. Nuclear containment comes with a variety of risks and issues, such as corrosion of steel containment shells in the drywell sand cushion region that results in wall thickness reduction to below the minimum allowable design thickness; corrosion of the steel containment shell;

Shots of a tethered ICM robot as it scales a windmill for blade inspection.

corrosion of the liner of the concrete containment with local degradation at many locations to approximately half the depth; leakage of the protective grease from the tendon sheathing of pre-stressed concrete containments; leaching and excessive cracking in concrete containments; and coating failure on containment structures.

The ICM climbing robots solve these issues by adhering to both steel and concrete, and climbing on vertical and inverted surfaces. The climbers are outfitted with UT/microwave/x-ray technologies that can measure and verify wall losses from corrosion, cracks, and coating degradation. The robots send the data to the operators, who then send it to offices anywhere in the world for a combined direction of the operation of the climbers during real time access to the readings. These kinds of measures and the identification of these issues is currently the only existing method to verify the integrity of the walls to the satisfaction of the NRC (Nuclear Regulatory Commission) and the utilities that need to ensure adequate safety measures.

Testing and ...

The ICM climbing robots handle non-destructive testing using infrared and radiological technologies. They can perform non-aggressive surface cleaning using water jets, cryogenics, and ice blasting. They can use cutting tools such as saws, scissors, and shears. They can be extended using mechanical arms, grippers, and materials-handling technologies. They can perform drilling and welding tasks.

Competing Climbing Technologies

The closest experimental competitors to ICM's climbers include the train wall bot which uses four legs and feet and a vacuum suction approach; the gecko inspired wall bot; the window cleaning robot called Urmakami; and the waalbot — none of which have the soft track, all surface, and angle scaling capabilities, or the adhesion capacity of the ICM robots. The waalbot climbs using rotating pads that stick to the wall and can surmount some corners, but did not have applications beyond this. A zero gravity car bot using suction would cling to and climb walls, but could not adhere with any real weight attached.

Another solution — the MagneBike — uses magnetic adhesion for power plant inspection. It cannot, however, adhere to surfaces other than metal ones. Another climber — the electro adhesive robot — applies electrostatic charges to walls through pads, to adhere to walls and climb them. This robot could actually climb concrete, metal, wood, and glass walls, though it could not maintain adherence if significant weight was applied.

Resource

International Climbing Machines climbing robots on Facebook
www.facebook.com/ICMClimbers

Forward Looking Thoughts

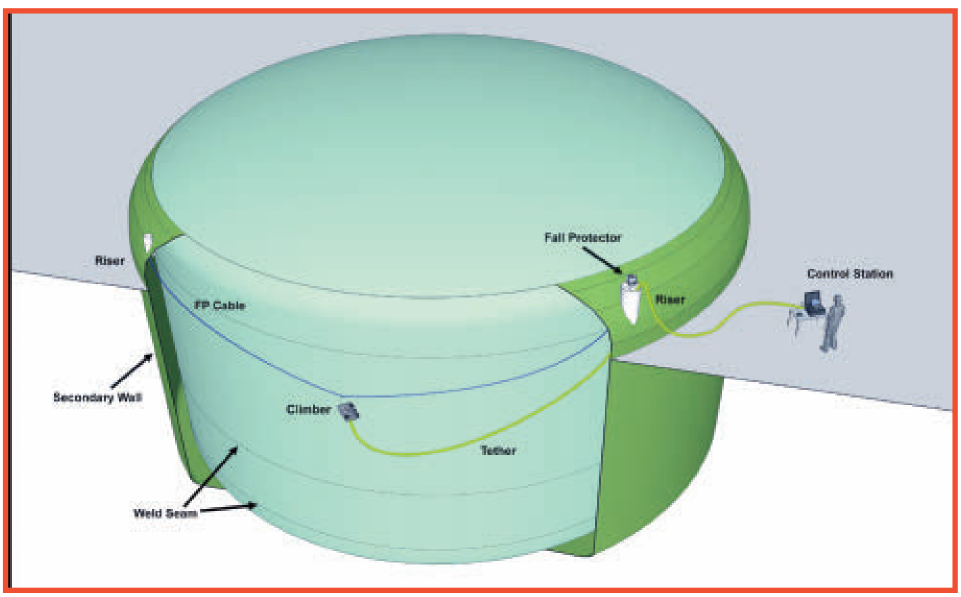
The International Climbing Machines bots can crawl up 300 foot wind turbine poles and are not blown off the sides in wind and weather. Though they have a wireless option, most are utilized using the tethered application.

The robots have replaced human turbine inspectors who must brake the turbine (stop it), and rotate the blades gradually while inspecting them with photography and telescopes from the ground. This older method can take hours, while the ICM robot takes much less time (minutes).

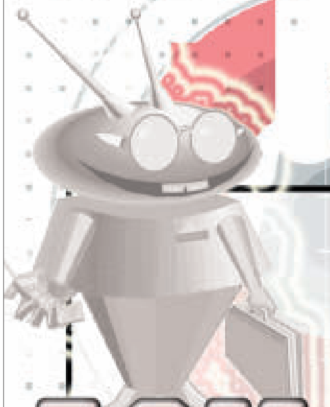
The wind turbine inspection approach is a collaboration of GE's Global Research Center and ICM applying a wireless video camera for inspections. GE plans to follow the video camera with a microwave technology that can look inside the wind turbines and poles.

Final Spin

The ICM robots are in use in real world industrial applications today, including inspecting large aircraft. There are many tasks that put humans in harm's way, and these climbing robots offer a safe solution. SV



Artist's rendering of the International Climbing Machines robot, as part of a tank wall cleaning system.



ASK MR. ROBOTO

Our resident expert on all things robotic is merely an email away.
roboto@servomagazine.com

Tap into the sum of *all human knowledge* and get your questions answered here! From software algorithms to material selection, Mr. Roboto strives to meet you where you are — and what more would you expect from a complex service droid?

by
Dennis Clark

How time passes. It's already getting to be time to make your (my) Santa's list for robot stuff we want but would never buy for ourselves. As I have mentioned recently, I spent some time at the Microchip Master's Conference in August, and saw many cool and inspiring things there. One of them was a chipKIT UNO32 wireless networking demonstration. This impressed me so much I talked the ear off of Gene Apperson (VP of Engineering) and Keith Vogel (Senior Software Engineer) about what they were doing. This was SO much fun that I've decided to tell you all about it. I've had questions in the past about networking a robot; this nifty demonstration shows how it can be done fairly inexpensively (more on that later). What I'm about to show you is a hands-on lab that Gene and Keith put on for a standing-room-only crowd at the conference.

As I've mentioned before, the Digilent chipKIT series of experimenter boards are designed to look like and work like the Arduino hobbyist boards. The UNO32 and MAX32 (and more recently, the uC32) have the same form factor as the Arduino UNO and MEGA boards. The chipKIT boards use an enhanced Arduino IDE called MPIDE (Multi-Platform IDE) that has the same look and feel as the Arduino, but adds the ability to utilize the Microchip PIC32 processor used on the chipKIT boards. Any Arduino program or library that isn't written for the AVR chip (by using AVR assembly code) will most likely run on a chipKIT board, as will any Arduino shield that can work at 3.3V. Okay, that was all the background. Now let's do some networking!

Installing the IDE, Sketch, and Libraries

If you haven't done so already, install the latest MPIDE from this site: **https://github.com/chipKIT32/chipKIT32-MAX/downloads**.

Pick your platform (all my screenshots will be from a Mac OS X machine). You will need an IDE of at least "20111221" — just get the latest. All MPIDE installs say they are Mpide-0023-(other stuff). The chipKIT Ethernet and Wi-Fi libraries require Java 6 (1.6.xx) to work, so make sure that you have them. I had to upgrade my OS to Snow Leopard for full compatibility. Once you have installed MPIDE, you are ready to get the *HttpServer* example sketch, C++ extensions, and chipKIT libraries.

My first exposure to MPIDE left me wondering where I should put libraries. For the chipKIT boards, you can't put them into the usual Arduino library location — that doesn't work (currently). Rick Anderson (one of the developers of the MPIDE environment) set me straight on this. You put chipKIT libraries in your sketch folder, in a folder called *libraries*. Then BAM! Everything works.

The demo code — available only here from yours truly — can be found at the article link. To install it, you first need to either find or create your default sketches folder. Fortunately, you can put this wherever you like; just set it up in the Mpide->*Preferences* window of MPIDE. Check out **Figure 1** to see what that looks like.

Before all of the web pages will work, you need to put the HTML files from the HttpServer/Content folder into the microSD card that you will install on the chipKIT Wi-Fi Shield. The system chipKIT demo code uses a FAT32 file system on the microSD formatted card like your PC uses on your hard drives, so if you mount the card on your PC you can just copy the files; microSD cards will almost always come with a full-sized SD card carrier that you can plug into your USB card reader (I use a Kodak 50-in-1 card reader). Do not format the card; it will work just as it comes from the store.

Now that you have Java 6 installed, MPIDE installed, and have selected your default sketches folder, you can unzip *HttpServer.zip* into your sketches folder. This zip file includes

both the demo sketch with helper code and the libraries in their correct locations.

Assembling the Hardware the HttpServer.pde Sketch Will Use

This code requires the following hardware:

chipKIT UNO32	(US $26.95)
chipKIT Wi-Fi Shield	(US $49.99)
chipKIT Basic I/O Shield	(US $37.99)
microSD Card (any size)	(US $5.99 for a 2 GB card)

Your project will probably not need the Basic I/O Shield, but this is a cool shield with a small OLED display, switches, buttons, LEDs, and more on it. It is very handy for prototyping. The microSD card is needed to hold the HTML pages that will be served up, and is located on the Wi-Fi Shield. **Figure 2** and **Figure 3** show a view of the individual boards and their final stack during use. The UNO32 is a 32-bit processor running at 80 MHz that has 128 KB Flash memory for programs and 16 KB RAM.

With everything assembled now, plug in the USB connection to the UNO32. The board will run off of the USB power, but an external 9V battery will allow you to put it anywhere or run it off of your robot battery.

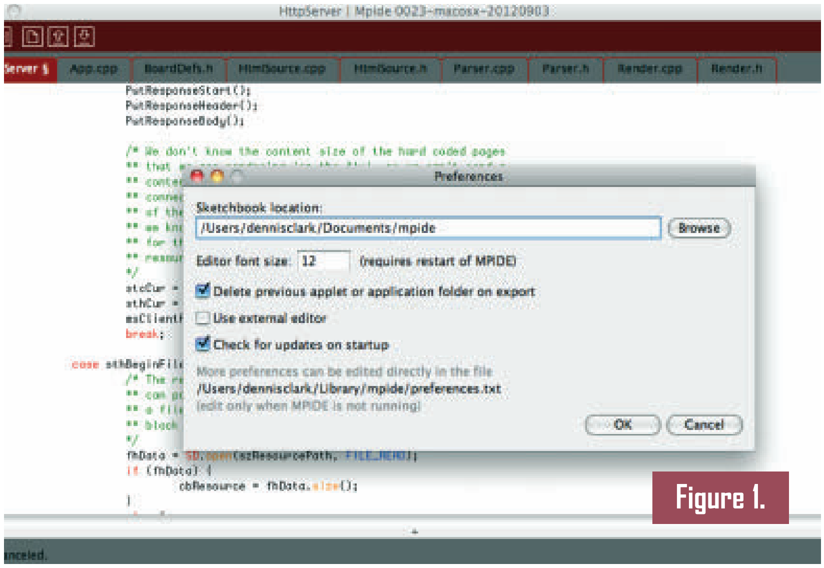


Figure 1.

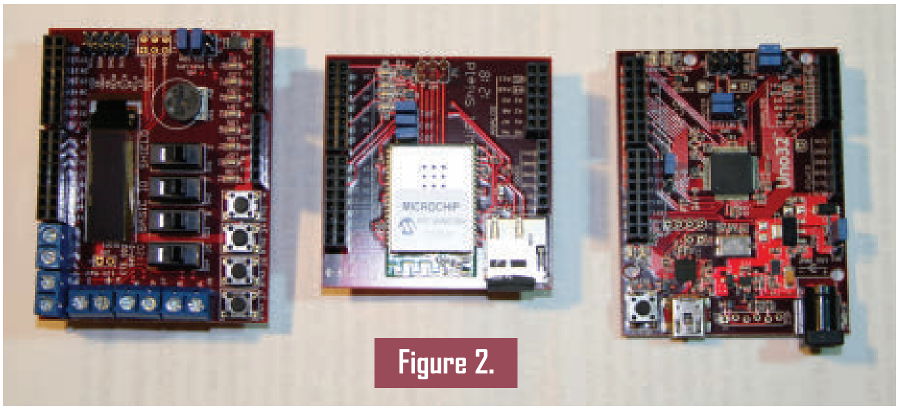


Figure 2.

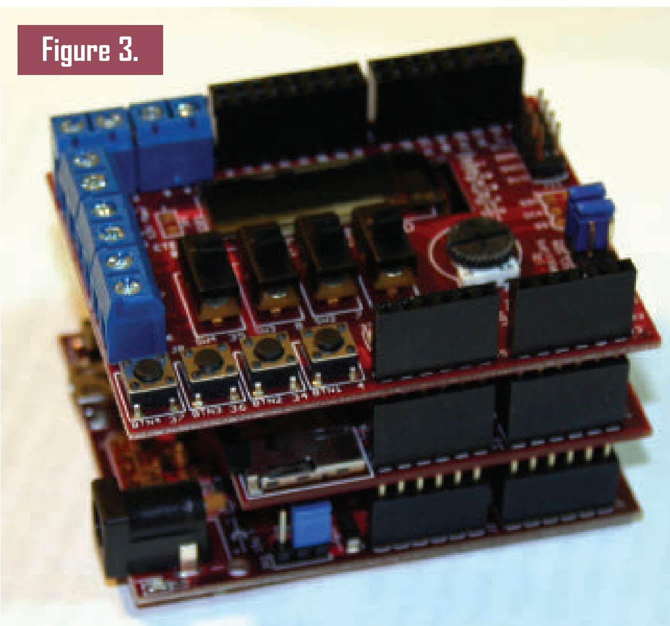


Figure 3.

Optional Hardware

My lab is too far away from my house's wireless router to work with this project, so I went to Best Buy and got a cheap (US $40) Cisco Linksys E1200 wireless router that I could fiddle with and not worry about the rest of the computers in the house (see **Figure 4**). So that this router didn't interfere with the Wi-Fi signal that my house normally uses, I changed its address by connecting a laptop directly to it. This made for a private wireless network that will come in handy for future experiments, as well.

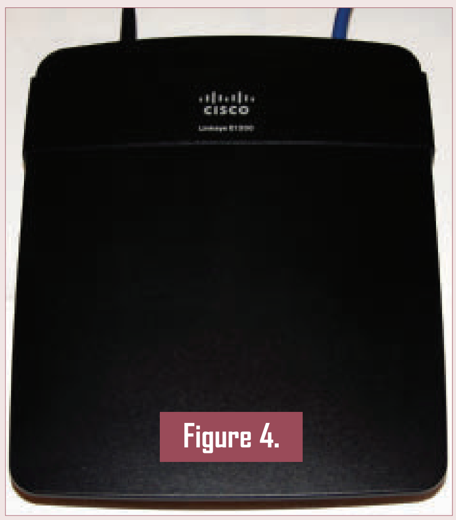


Figure 4.

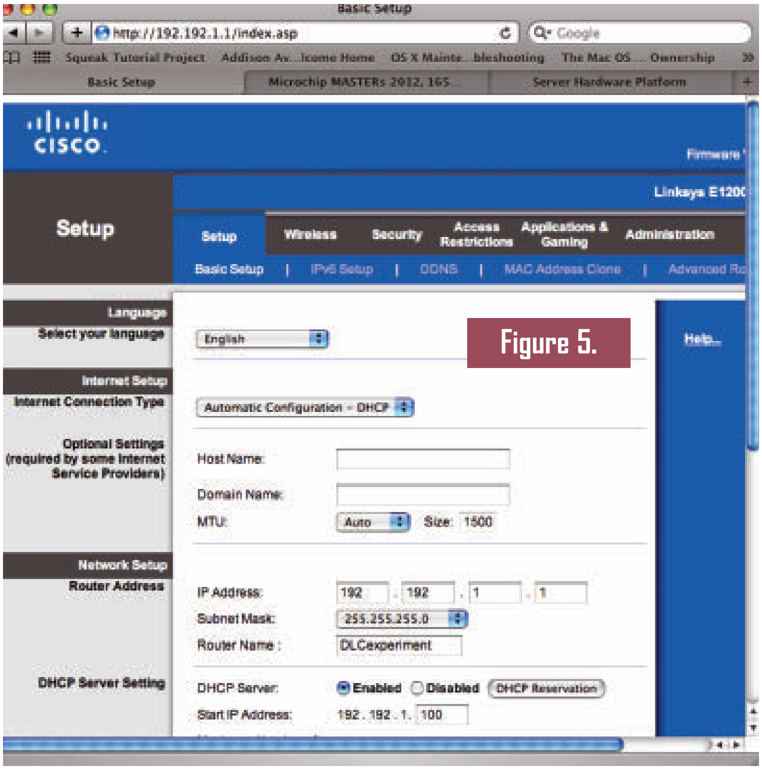


Figure 5.

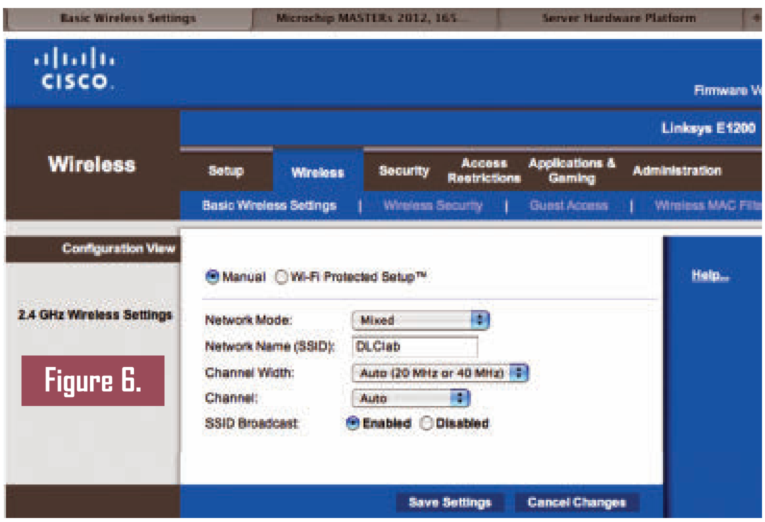


Figure 6.

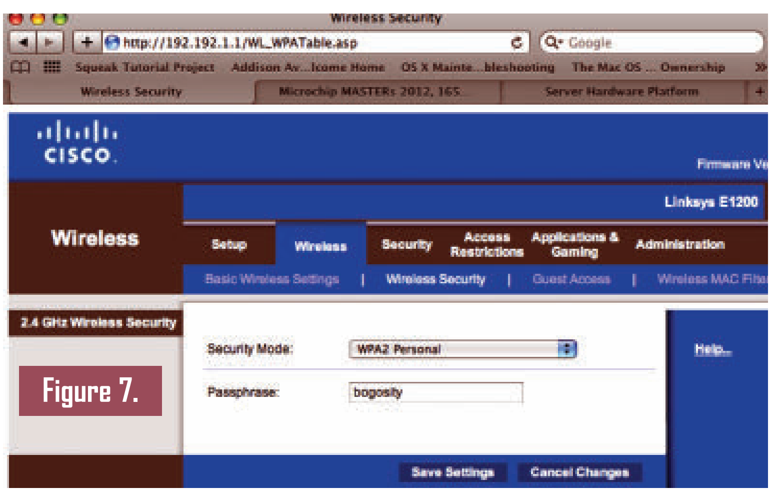


Figure 7.

Step 1: Set up a private router network.

Let me walk you through futzing with a Linksys router to get set up for this project. First off, you can skip this first step if you only have one wireless router and aren't creating your own private robot network. If you are like me, read on.

Plug your laptop's CAT5 cable directly into Slot 1, NOT the Internet jack on the router. Now, go to your router's admin pages — typically at 192.168.1.1 — and log in as (typically) admin, with the password "admin." Navigate to the Setup tab and the Basic Setup sub-tab (**Figure 5**).

Change your IP Address to something you like; I chose 192.192.1.1. Name your router if you wish (why not?). You will want the DHCP server to be enabled; that is what our UNO32 is going to use. Save your changes and reboot (power cycle) your router. You may need to unplug and re-plug in your CAT5 connection to the laptop.

Step 2: Configure your wireless settings and WPA2 passkey.

Now, you need wireless to be set up to be compatible with your project. Navigate to the Wireless tab and the Basic Wireless Settings sub-tab (**Figure 6**). Set your Network Name (SSID); I chose "DLClab." Enable SSID Broadcast because the chipKIT libraries need it. Use the "Manual" mode. Save your settings.

Now, navigate to the Wireless Security sub-tab; set your Security Mode to "WPA2 Personal," and your Passphrase to something; I chose "bogosity" (**Figure 7**). Save each of your changes. You are now ready to configure your sketch and play with your boards!

Customizing the HttpServer Sketch

You will need to customize the HttpServer sketch to match your Wi-Fi and network settings. Don't worry! This is easy. If you have installed all the files as I've instructed, you can now open your sketch by File->Sketchbook->HttpServer. You will get an MPIDE screen with a bunch of files in tabs as shown in **Figure 8**.

We need to edit HttpServer.pde to use the router and Wi-Fi settings you entered in Step 2 above. Look for the following lines in the sketch and set them as below:

```
// Set to 1 if using open security,
// otherwise WPA2 is used
#define USE_OPEN_SECURITY    0
// Only has meaning if doing WiFi, 0 ->
WPA2, 1 -> Open
```

```
// Set to 1 if using DHCP, 0
// if using a static IP
#define USE_DHCP     1
// 0 -> Static IP, 1 -> DHCP
```

Now, look for the following two lines in the sketch and set them to the SSID and Passphrase that you configured your router to use in Step 2:

```
char * szSSID      = "DLClab";
// the name of the network to
// connect to
char * szPassPhrase  =
"bogosity";
// pass phrase to use with
// WPA2
```

You can search the sketch using "***DLC" if you like. I put those comments into the code so that I could find them again later. You are now ready to compile and download your sketch.

To compile and download, press the box with an arrow pointing to the right in the upper left of the MPIDE window (refer again to **Figure 8**). After it finishes crunching data and downloading to your board, press the Serial Monitor button in the upper right of the MPIDE window. After about 40 seconds, you'll see the serial port output shown in **Figure 9**. It typically takes about 40 seconds.

This all means that you now have a connection to your wireless network. Pat yourself on the back for your tireless efforts! If you look at your router, you'll see your board listed in the DHCP client table which you will find by navigating to the "Status" tab and the "Router" sub-tab, and clicking on the "DHCP Client Table" button (**Figure 10**).

This router table is another place that you can confirm your connection status while troubleshooting your work.

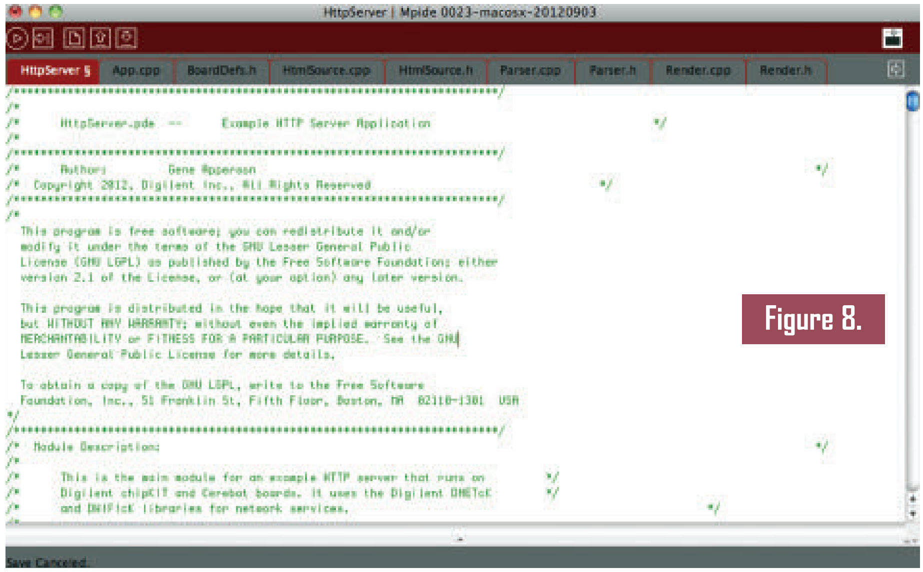


Figure 8.

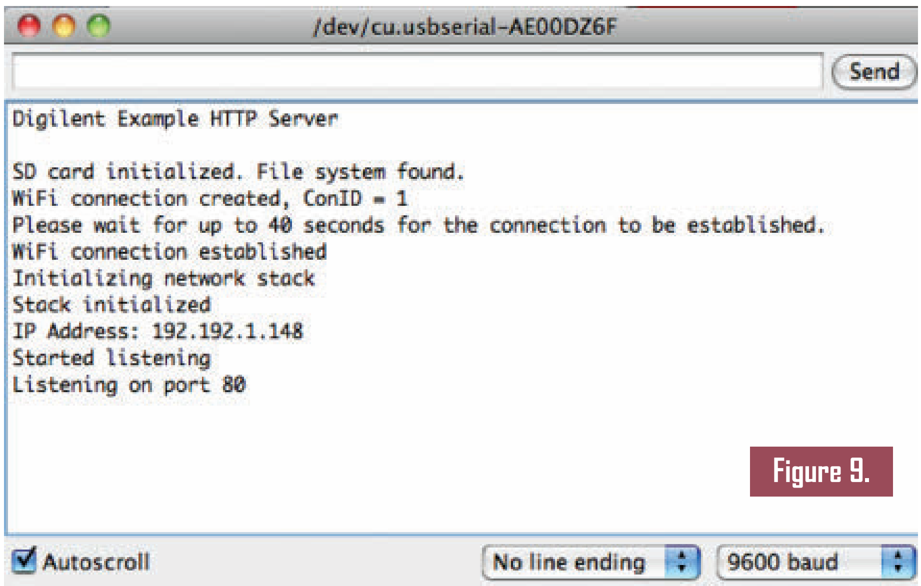


Figure 9.

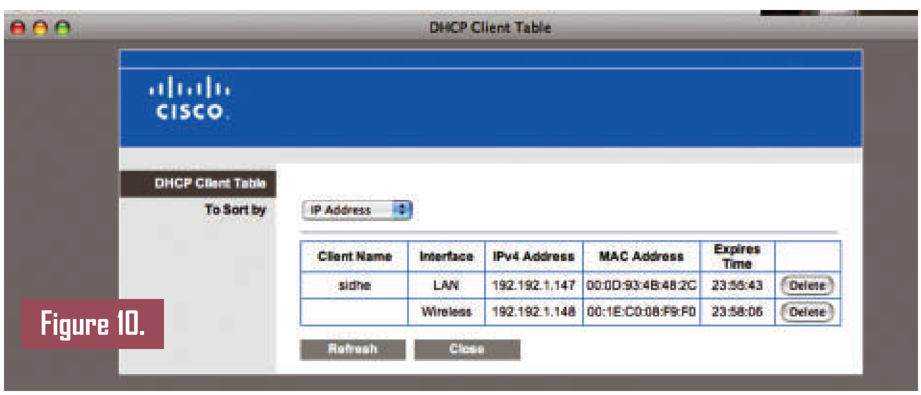


Figure 10.

Playing With Your New Toy

Now that you are running and on the network, let's talk to the board! **Figure 11** shows the web page that you will get when you use your favorite browser and type in the IP address given in **Figures 9** and **10**. Before you worry about it, there is a bit of a glitch in this setup.

Some of the button and switch I/O lines are being used by the Wi-Fi

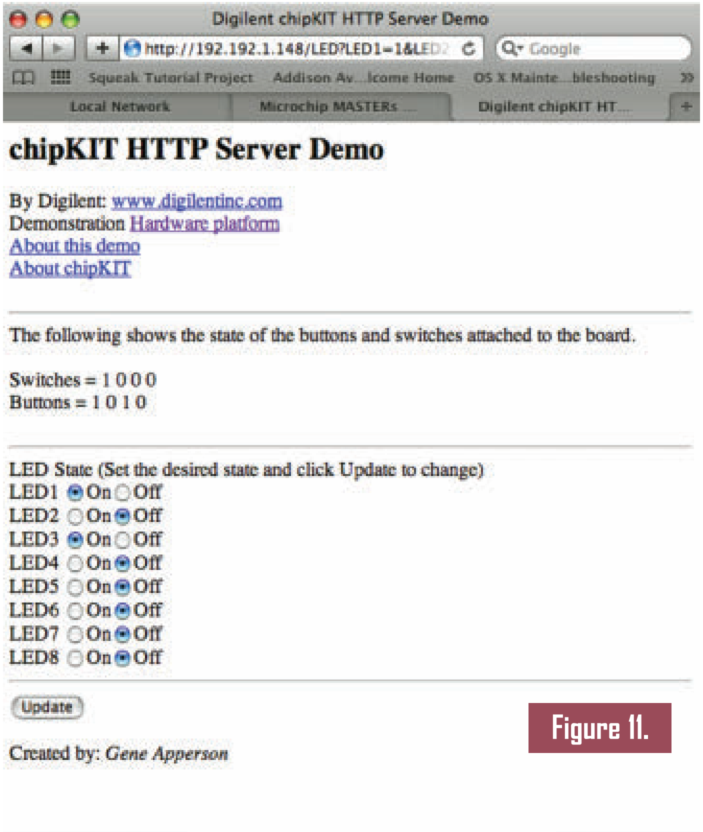


Figure 11.

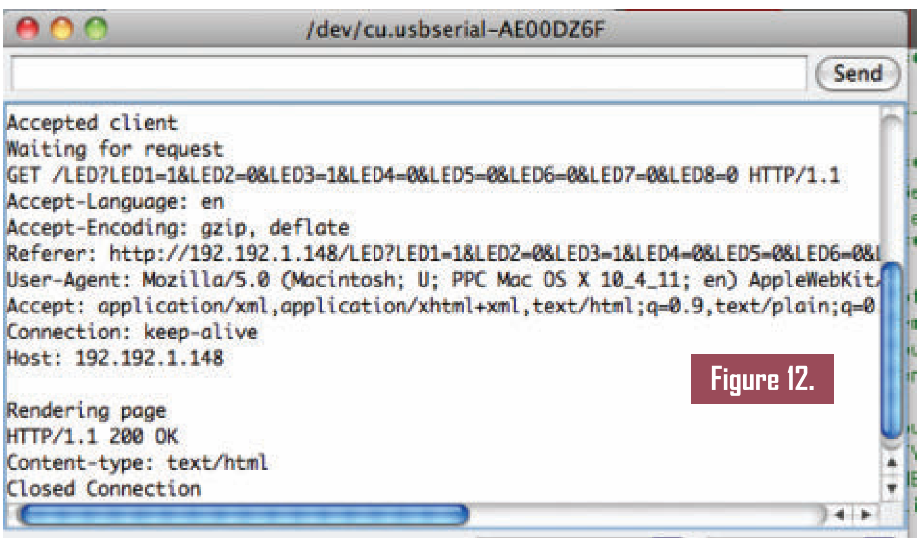


Figure 12.

Shield, so they can't be changed on the I/O board.

That is why you see some of the buttons and switches on the web page set to a '1.' Experiment now with switch settings and buttons, hit the "Update" button on the page, and see them change.

Your robot can also give you sensor feedback. Click on the radio buttons for the LEDs, then "Update." You will see them change on the I/O board. In my case, the web page I show causes the LEDs to light on the board; look and see how they are mapped.

Figure 12 shows the serial port output telling you the HTML messages are coming in from the client web page. This is handy for troubleshooting and learning how HTML works. Since this is a web server, more than just this can be done. If you click on the "Demonstration Hardware platform" link on the page, you'll get **Figure 13** with a graphic on it. This could be your robot sending back camera shots. Cool, huh?

Setting the LEDs can also be you sending instructions to your robot over the web. Now we have telepresence with our robot, as well. Sure, this can all be done with a Linux or Windows machine on your robot and lots of programming, but with this example code, you are doing that at a fraction of the cost and possibly a fraction of the time.

But wait! There's more! You don't need to lug around your laptop to send commands to your robot. You can use your handy smartphone, and don't need to write a single app to do it. **Figure 14** shows my iPhone interacting with the UNO32.

The majority of the interaction with the UNO32 Wi-Fi project is done in the C++ files: HtmlSource.cpp, Parser.cpp, and Render.cpp. The HttpServer.pde file just sets up the actual HTTP server.

This can be re-written to give more Arduino-esque handling of the interaction with the outside world, but this is a great start with well written example code that can get us up and running fast. I plan on playing with this more in the future, so stay tuned!

I hope this has given you some ideas; it sure has given me some. I'm visualizing a CMU-CAM interface taking pictures to send back via Wi-Fi or giving directions to the robot from a remote location over a website from across the world.

Hopefully, you have gotten some good ideas too from this writeup. As usual, you can reach me with any questions that you might have at roboto@servomagazine.com.

I'll be happy to work on them! Until next time, keep on building those robots! **SV**

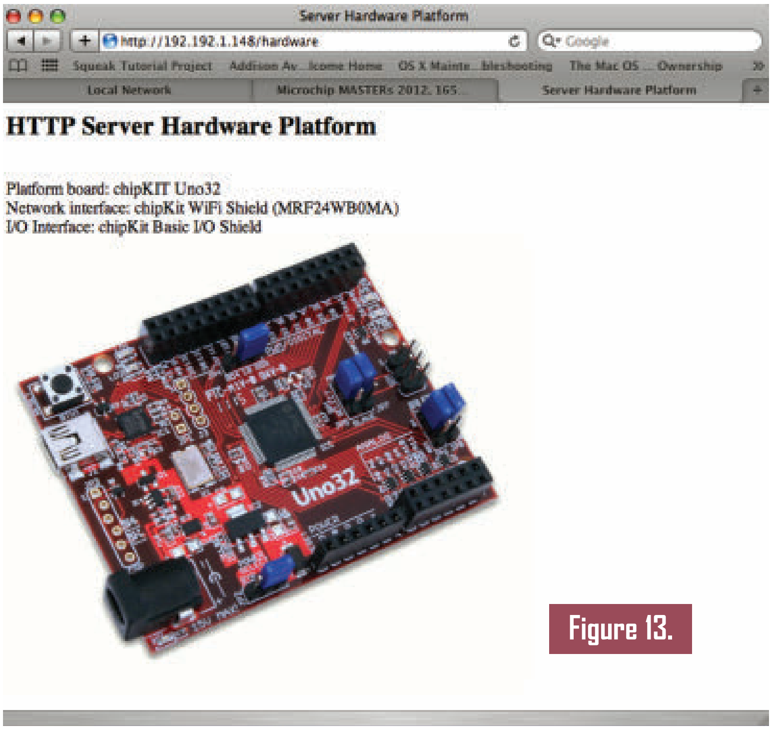


Figure 13.

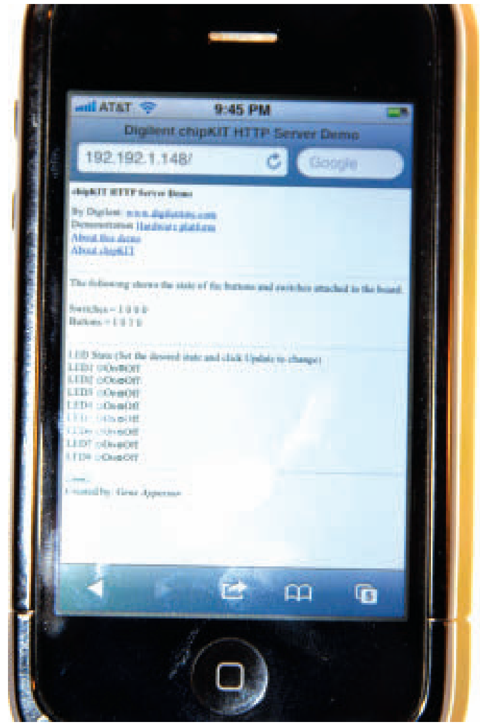


Figure 14.

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com

EVENTS Calendar

ROBOTS.NET

Send updates, new listings, corrections, complaints, and suggestions to: steve@ncc.com or FAX 972-404-0269

Know of any robot competitions I've missed? Is your local school or robot group planning a contest? Send an email to steve@ncc.com and tell me about it. Be sure to include the date and location of your contest. If you have a website with contest info, send along the URL as well, so we can tell everyone else about it.

For last-minute updates and changes, you can always find the most recent version of the Robot Competition FAQ at Robots.net: **http://robots.net/rcfaq.html**.

— R. Steven Rainwater

NOVEMBER

3 **Atlanta Hobby Robot Club (AHRC) Robot Rally**
Pinckneyville Community Center, Norcross, GA
Events include line maze solving, mini Sumo, and the Robot Polyathlon. The polyathlon is made up of six individual contests: Simple Line Follower, Advanced Line Follower, Beacon Killer, Beacon Killer with Obstacles, Navigation by Dead Reckoning, and Bulldozer.
www.botlanta.org/robot-rally

3 **Bloomington VEX Tournament**
Bloomington, IN
This year's Boomington Robotics Club events include Light the Lamp, Full Pull XL, and F8 Autonomous.
https://sites.google.com/site/bloomingtonroboticsclub

4 **International Micro Robot Maze Contest**
Noyori Memorial Hall, Nagoya University, Japan
While the big attraction is the 1 cm cube micro robot maze event, there are four other events including 1 cm teleoperated robot maze, 1 inch robot maze, legged micro robots, and best free style performance by a micro robot.
http://imd.eng.kagawa-u.ac.jp/maze/events.html

23-25 **All Japan Micromouse Contest**
Toyosu, Koto-ku, Japan
Events for autonomous robots including classic Micromouse, half-size Micromouse, and robot race.
www.ntf.or.jp/mouse

23-25 **Robotex**
Tallinn, Estonia
This is the largest autonomous robot competition in Estonia. This year's events include robot football, line following, mini Sumo, and LEGO Sumo.
www.robotex.ee

25 **Robocon**
Tokyo, Japan
Student teams from all over Japan come together at Robocon, where the robots they've designed compete in the Robo Bowl.
www.official-robocon.com

DECEMBER

1-2 **South's BEST Competition**
Auburn University, Auburn, AL
Regional for the BEST student competition.
www.southsbest.org/site

6 **Roboexotica**
Vienna, Austria
This annual competition of cocktail robots includes events for best cocktail service, best cocktail mixing, and best bartending conversation.
www.roboexotica.org

15 **Robotic Arena**
Wroclaw, Poland
Lots of events including mini Sumo, micro Sumo, nano Sumo, Micromouse, line following, and freestyle.
www.roboticarena.org

NEW PRODUCTS

Four-Motor Controller System

The new four-motor controller system from ServoCity allows you to proportionally control the speed and direction of up to four motors with a dual joystick controller. The four-channel speed controller converts the PWM input from the dual joystick into varying voltages which travel through the CAT6 cable (sold separately) to the CAT6 motor circuit board where the motor wires can be soldered in. This system will provide fine-control for any of ServoCity's compatible gearmotors.

This is an open loop controller which means that the further the joystick is deflected from the center position, the faster the motors will run. When the joysticks are released, the motors will stop and hold their position until another input is received. The four-motor controller system is perfect for pan and tilts, time-lapse rigs, camera sliders, or any application which requires fluid bi-directional proportional control of a DC motor.

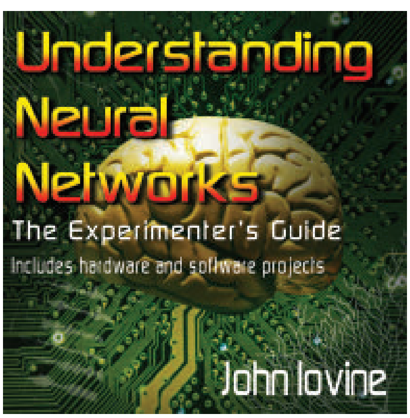
Smooth Hub Pulleys

ServoCity is also offering a full line of smooth hub pulleys. These hub pulleys are machined from 6061 T-6 aluminum with a 1/2" center bore and a 1/4" face width. The interior face of the larger pulleys has been machined down to 1/8" to save on unnecessary weight, and allow the pulley to be mounted on either face to change the offset. The hub pulleys are offered in 1", 2", and 3" face diameter, and contain the .770" ServoCity hub pattern. ServoCity carries a full line of .770 hubs for servos and for various sizes of shafting. The new pulleys will work with both 1/4" and 1/8" smooth belts. These pulleys are also ideal for a variety of applications, such as camera sliders, time-lapse setups, robotic applications, and more.

For further information, please contact:

ServoCity	Website: **www.servocity.com**

Book On Artificial Neural Networks



Understanding Neural Networks is an introductory book written by John Iovine about artificial neural networks. The book begins with examining biological neurons in the human brain and defining their real world mathematical and electronic equivalent.

Building upon this foundation, the text contains hardware and software projects that illustrate neural networks. Hardware projects include an op-amp neuron that tracks a light source, a speech recognition system, and machine vision system.

Software projects include a preceptron program and back-propagation networks. The book provides a comprehensive introduction to neural networks.

In *Understanding Neural Networks*, Iovine also explains the differences between traditional

rule-based (symbolic) computer processors and the oftentimes mind-boggling possibilities of neural networks (artificial intelligence). Following the introductory explanation of the science and history of development, Iovine delves deeper into the subject, covering additional topics such as: biological and mathematical neurons, a neural network speech recognition circuit, neural paradigms, back propagation, teaching computers to speak, computers that smell fuzzy logic, plus more.

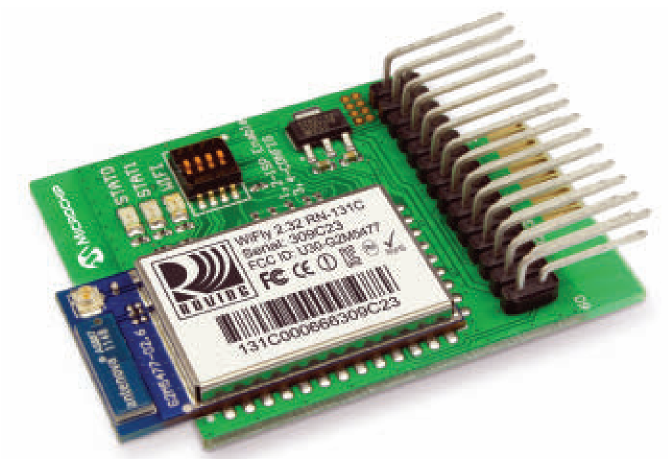
If these subjects intrigue you, *Understanding Neural Networks* will help you understand the nuts-and-bolts of neural networks, along with the whys, hows, and what-ifs.

For further information, please contact:

Images, Co. Website: **www.imagesco.com/servo/pssmc-m.html**

New Embedded Wi-Fi® Development Boards

Microchip Technology, Inc., announces the integration of its Wi-Fi® modules from the recent Roving Networks acquisition into its flexible, modular Explorer development systems supporting all of Microchip's eight-, 16-, and 32-bit PIC® microcontrollers. The RN-131 and RN-171 PICtail™/PICtail Plus daughterboards are the first



two products developed by Microchip based on Roving Networks modules.

These modules use a simple serial interface to connect with any PIC, and expand Microchip's wireless portfolio with the industry's lowest power consumption along with an integrated TCP/IP stack in a certified Wi-Fi solution.

The Roving Networks RN-171 and RN-131 fully certified modules from Microchip are comprehensive networking solutions that include a true 802.11 b/g radio, baseband processor, TCP/IP stack, and a host of networking application features. No external processor drivers are required, enabling Wi-Fi connectivity for four-, eight-, 16-, and 32-bit processors. This onboard stack approach significantly reduces customer's integration time and development efforts in a small form factor, while offering ultra-low power consumption (down to 4 µA in sleep, 35 mA in receiver, and 120 mA in transmit mode).

The RN-131 PICtail daughterboard is available for $44.95 each; the RN-171 PICtail daughterboard is available for $39.95 each.

For further information, please contact:

Microchip Technology, Inc. Website: **www.microchip.com**

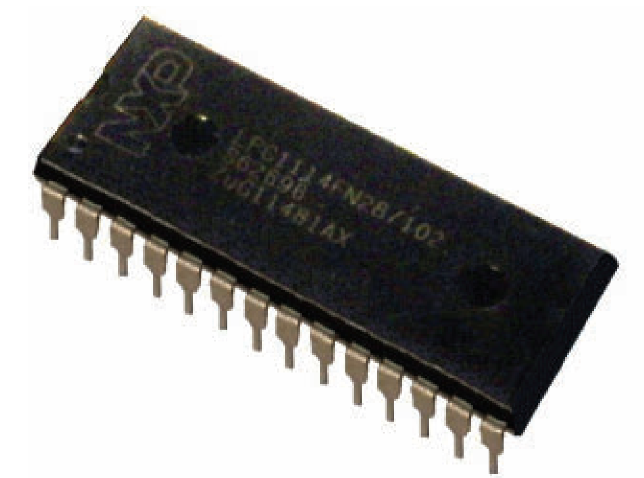
$10 System On a Chip

The ARM BASIC Chip from Coridium Corp is a system on a chip with IEEE floating point support for $10. The ARM BASIC Chip is ideal for:

- Students learning how to program and develop applications.
- Educators and reseachers designing prototypes.
- DIYers exploring robotics, wireless, and sensors.
- Factory automation and data collection.

Technical specifications include:

- 50 MHz ARM M0 32-bit CPU (LPC1114).
- Programmable in a compiled BASIC.
- Operates faster than 10 million BASIC lines per second.
- IEEE 754 floating point support.
- Internal 12 MHz 1% oscillator.
- DIP28 - 1.4′ x .06″ size.
- Storage of 32K Flash memory; 4K RAM memory; 20K user Flash space; and 2K user RAM.
- Power is less than 50 mW; 3.3V DC input; 22 TTL compatible; 2.4V threshold, 5V tolerant.
- Connections for 22 digital I/O.



The ARM BASIC Chip is easy to use and program in compiled BASIC. Check out the websites for a comprehensive description of the product's features, benefits, and technical specifications.

For further information, please contact:

Coridium Corp. Website: **www.basicchip.com** or **www.coridiumcorp.com**

Toolbox Size Battery Tester

Saelig Company, Inc., has introduced the BT-168D battery tester which is an ideal toolbox companion for reading the output voltage level of any 1.5V or 9V battery. Featuring sturdy construction and an easy-to-read digital LCD display, the BT-168D has an adjustable sliding arm for contacting the terminals of different sizes of batteries, even miniature coin cells. Its small size makes it suitable for an engineer's traveling toolkit.

Features include:

- Small size for easy use and portability.
- Tests AAA, AA, C, and D type 1.5V batteries.
- Tests 1.5V miniature button cells.
- Tests 9V MN1604 type batteries.
- Suitable for standard and rechargeable batteries.
- Built-in load resistor for accurate results.
- Auto polarity indication.
- Non-polarized testing terminal.
- Weight of 3 oz.
- Size: 7.3" x 4.3" x 1.4".

The BT-168D battery tester is available now for $11.95. For further information, please contact:

Saelig	Website: **www.saelig.com**

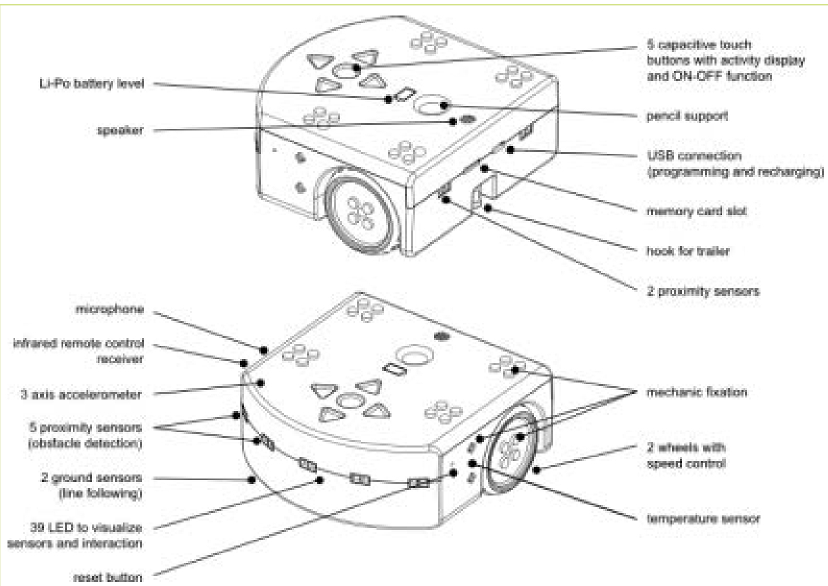
Is your product innovative, less expensive, more functional, or just plain cool? If you have a new product that you would like us to run in our *New Products* section, please email a short description (300-500 words) and a photo of your product to: newproducts@servomagazine.com.

bots IN BRIEF

THYMIO A GREAT BUYMIO

How much robot can you get for a hundred bucks? Not much, if you consider what commercial robotic vacuums go for. However, you could spend it on an open source education robot from Switzerland that will help your kids to learn how to do things besides not vacuuming the floor.

The Thymio II comes from those robot geniuses at EPFL Lausanne in Switzerland. It's an educational robot designed from the ground up to be easy and fun to mess with for people with very limited (or no) previous experience in robotics. It's also designed from the ground up to be cheap, at just about $100 USD. How is this possible, you ask? Apparently, there's pretty much no profit margin or distribution cost, and all you're paying for is the hardware and for some people to put it together for you into a working robot. Not too shabby. Plus, this robot comes with a bunch of sensors and other goodies. Interactivity is based on several functionalities:



- Capacitive touch buttons.
- Color of the body (full RGB spectrum).
- LED associated with each robot function.

Yes, there's a trailer hook, so you can stop worrying about that, and those "mechanic fixation" points are LEGO compatible.

To program Thymio II, you can use a nifty graphical interface, or a simple programming language called Aseba that's similar to Matlab.

Actually, $100 seems very cheap for a platform like this — cheap enough that a $1,000 grant could outfit an entire classroom with robots that are colorful, versatile, fun, and can be tackled with a GUI before graduating to writing code. There's lots more info along with examples of what Thymio II can do at **https://aseba.wikidot.com/en:thymio**.

iROBOT IN MINT CONDITION

iRobot has just announced its acquisition of Evolution Robotics — maker of the Mint swifferbots — for $74 million in cash.

If you're not familiar with Mint, it's a robot that sweeps or mops using Swiffer pads. It's not a vacuum, and it only works on hard floors. It's quiet, though, and cheap at just $200 for the base model. It uses a beacon system to localize itself, meaning that it can sweep in straight lines instead of in a pseudo-random motion like the Roomba. Whether or not straight lines are better for cleaning robots is debatable, but localization has the potential to enable all kinds of clever new robots. This seems to be a big part of why iRobot made the buy, since Evolution also has patents on an image-based localization and mapping system called vSLAM that would be appropriate for small mobile consumer robots.

bots IN BRIEF

HEARD IT THROUGH THE GRAPEVINES

A little robot named Wall-Ye is trying to get involved in the wine business process from the ground up by helping out in vineyards in France.

Wall-Ye has two arms and six cameras, weighs 20 kilos, and can reportedly autonomously prune 600 vines per day — among other things — according to a report.

Wall-Ye draws on tracking technology, artificial intelligence, and mapping to move from vine to vine, recognize plant features, capture and record data, memorize each vine, synchronize six cameras, and guide its arms to wield tools.

Yours for just $32,000.

To prune a vine, you first need a robot that can reach the vine (many if not most are trellised fairly high up), and second, you need a fairly complex vision system to be able to map the vines in 3D with sufficient accuracy and precision to properly cut them.

So, it seems like Wall-Ye may not be able to do this easily. But again, Wall-Ye doesn't *have* to be a pruning robot. It would be quite valuable as a mobile monitoring system with the capability to measure temperature, moisture levels, soil PH, and (if you wanted to get fancy about it) even vigor levels and vine health with a hyperspectral sensor using technology. Just a helpful suggestion.

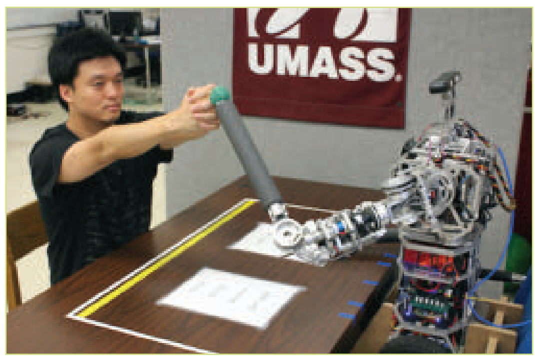
ROBO-THERAPY

Researchers in robotics and communication disorders at the University of Massachusetts Amherst have teamed up to explore whether a personal humanoid robot may help people recovering from stroke by delivering therapy such as word-retrieval games and arm movement tasks in an enjoyable and engaging way.

Speech language pathologist Yu-kyong Choe recently won a two year $109,251 grant from the American Heart Association to investigate the effect of stroke rehabilitation delivered by a humanoid robot — a child-sized unit with arms and a screen where therapists, doctors, and others can interact with a client. Choe is collaborating with Rod Grupen, director of UMass Amherst's Laboratory for Perceptual Robotics, on ways to bring more and longer-term, home-based therapy and social contact to people recovering from stroke.

The study will enroll five stroke patients per year to attend three sessions per week for five weeks at the UMass Amherst lab. Three treatments will be compared: computer-mediated, robot-mediated, and robot-assisted telepractice by a remote therapist.

In the robot-mediated condition, patients complete word-retrieval tasks and games, plus arm exercises delivered by the robot alone based on therapy routines it has observed. In the computer-mediated condition, the same tasks and exercises will be presented on a laptop computer.



In the robot-assisted telepractice condition, the client performs word- and arm-movement tasks designed and directed by a therapist in a remote location being observed and mimicked by the robot via a 3D range camera. The robot exactly copies the therapist's movements. Choe predicts that the two robot-mediated conditions will yield better outcomes in both speech and physical function because of the patient-robot interactions. The research team will also analyze how the telepractice and robot-mediated therapy sessions are received by client and therapist.

Therapists Jennifer Baird and Tammie Foster, with computer science doctoral students Takeshi Takahashi and Hee-tae Jung are working with patients three days per week and developing software for uBot5 — the adaptive humanoid robot — to act as the liaison between a remote therapist and the client at home.

SUPPORT YOU LOCAL HUMAN

With virtually zero warning, Toyota has introduced their Human Support Robot, or HSR. Designed to assist disabled people, the HSR looks like it might be good for all kinds of household tasks whether you're disabled or not.

There wasn't a lot of information beyond what Toyota provided in a recent release at press time, but what we know is the HSR has a tablet for a head that can be used for telepresence or as a graphical interface (as with Ava). The base is mobile, with a PR2-style articulated torso that can extend from about half a meter in height up to just under a meter and a half. It sort of looks like there's an Xtion in the head, but we're not sure what other sensors might be in there as well. Top speed is 3 kph. The HSR can make it over 9 mm obstacles and climb slopes of five degrees.

The arm — which appears to have six or seven degrees of freedom plus the gripper — can lift small objects that weigh 1.2 kilos or less. It's designed to move slowly enough to be inherently safe, but it's not clear yet what sort of precision or intelligence is built into it.

No info on price yet either.

LETTUCE HELP YOU

Blue River Technology continues to work on a weeding bot. Instead of using pesticides, Lettuce Bot uses cameras, computer vision, and algorithms to discern vegetables from weeds, and the company claim is that they have 98-99% accuracy. It then spreads fertilizer down to kill the unwanted plants.

The company has made two prototypes, and the ultimate goal is to make an organic weeder which will undoubtedly happen in the near future, since they have funding from the National Science Foundation.

Weed elimination — a necessity for agriculture — currently uses chemical herbicides resulting in health and environmental problems. So, we want to replace hazardous chemicals with robotic technology. Over $25 billion is spent per year in herbicides. It's a good thing to replace hazardous chemicals with robotic technology.

Blue River is working with leading experts in this area including professors from UC Davis and Stanford, as well as leading crop producers.

THE OWNERSHIP, MANAGEMENT, AND CIRCULATION STATEMENT OF SERVO MAGAZINE, Publication Number: 1546-0592 is published monthly. Subscription price is $24.95. 7. The complete mailing address of known office of Publication is T&L Publications, Inc., 430 Princeland Ct., Corona, Riverside County, CA 92879-1300. Contact Person: Larry Lemieux. Telephone: (951) 371-8497. 8. Complete Mailing address of Headquarters or General Business Office of Publisher is T&L Publications, Inc., 430 Princeland Ct, Corona, CA 92879. 9. The names and addresses of the Publisher, and Associate Publisher are: Publisher, Larry Lemieux, 430 Princeland Ct., Corona, CA. 92879; Associate Publisher, Robin Lemieux, 430 Princeland Ct., Corona, CA 92879. 10. The names and addresses of stockholders holding one percent or more of the total amount of stock are: John Lemieux, 430 Princeland Ct., Corona, CA 92879; Lawrence Lemieux, 430 Princeland Ct., Corona, CA 92879; Audrey Lemieux, 430 Princeland Ct., Corona, CA 92879. 11. Known Bondholders, Mortgagees, and other security holders: None. 12. Tax Status: Has not changed during preceding 12 months. 13. Publication Title: Nuts and Volts 14. Issue Date for Circulation Data: October 2011-September 2012. 15. The average number of copies of each issue during the proceeding twelve months is: A) Total number of copies printed (net press run); 10,248 B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 4,281 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 2,048 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 6,330; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 808; E) Total Free or Nominal Rate Distribution: 808; F) Total Distribution: 7,138; G) Copies not distributed: 3,110; H) Total: 10,248; Percent paid circulation: 88.68%. Actual number of copies of the single issue published nearest the filing date is September 2011; A) Total number of copies printed (net press run) 9,449; B) Paid/Requested Circulation (1) Mailed Outside County subscriptions: 4,126 (2) Mailed In-County subscriptions: 0 (3) Paid Distribution Outside the Mail including Sales through dealers and carriers, street vendor, and counter sales and other paid distribution outside USPS: 2,051 (4) Paid Distribution by other classes of mail through the USPS: 0; C) Total Paid Distribution: 6,117; D) Free or Nominal Rate Distribution by mail and outside the mail (1) Free or Nominal Rate Outside-County Copies: 0 (2) Free or Nominal Rate In-County Copies: 0 (3) Free or Nominal Rate Copies Mailed at other classes through the USPS: 0 (4) Free or Nominal Rate Distribution Outside the mail: 300; E) Total Free or Nominal Rate Distribution: 300; F) Total Distribution: 6,477; G) Copies not distributed: 2,972; H) Total: 9,449; Percent paid circulation: 95.37%. I certify that these statements are correct and complete. Lawrence Lemieux, Publisher - 10/04/2012.

STARTLED INTO ACTION

A biologically-inspired threat monitoring system called STARTLE detects anomalous or threatening conditions by emulating the mammalian conditioned-fear response mechanism. It provides a mechanism of cued-attention which has significant performance and efficiency benefits over conventional techniques.

Applications for STARTLE include:

Autonomous Vehicles

STARTLE can provide enhanced situational awareness and an early threat warning to both an autonomous vehicle and its remote operator. Making use of existing hardware, STARTLE intelligently processes information from multiple onboard sensors, cueing systems to assess and confirm potential threats to the vehicle.

Health and Usage Monitoring Systems (HUMS)

The threat detection methods in STARTLE also make it ideal for the detection of anomalous subsystem conditions within a complex system; for example, in internal system status monitoring of manned or autonomous platforms.

Computer Network Defense

STARTLE can be used for the detection of sophisticated computer network threats. It has been demonstrated successfully in detecting penetration test activity in real world computer networks, and can use intercept-derived data from local and cloud-based storage.

Benefits of STARTLE include: autonomously detects and characterizes threats; automatically directs sensor assets; processing power freed up for other tasks; plus it reduces operator workload.

Key components are:

Neural network – Gives high performance, lightweight monitoring and alerting.

Rule system — Compiled by domain experts, and allows the system to request the most appropriate data to confirm potential threats. Cued sensor re-tasking obtains additional data to be collected from a particular sensor/data store/processing algorithm and gives traceability which allows reasoning to be validated.

Neural network training can be based on real or synthetic environment derived data, allowing a wide range of potential operational scenarios to be investigated

Learning — Potential to accommodate on-the-job learning in future systems.

Cool tidbits herein provided by www.botjunkie.com, www.robotsnob.com, www.plasticpals.com, http://www.robots-dreams.com, and other places.

COMBAT ZONE

Discuss this section in the *SERVO Magazine* forums at http://forum.servomagazine.com

Featured This Month:

www.servomagazine.com/index.php?/magazine/article/november2012_CombatZone

BUILD REPORT:

Algos: Who Needs MOI When You've Got Angular Velocity?

● by Michael Jeffries

After competing for a year with a two-wheel drive wedge named Kobalos, I decided to retire that robot and move on to something with an active weapon. During the initial concept stage of Algos, I decided on a few key design elements that had to be present. First, I wanted to build Algos primarily out of titanium. Second, I wanted to try my hand at designing an effective single tooth weapon. Third, I wanted to have a secondary attack method should the weapon fail. After sketching several concepts, I settled on a two-wheel drive robot with a wedged front and a high rpm small diameter spinning disk.

This design utilizes many of the same components used in Kobalos and many of my other 1 lb robots. For the drive, I'm using the same 11:1 Fingertech Silver Spark gearboxes that Kobalos has used for nearly a year that have (as of now) survived 22 fights and hours of testing. I'm also using Fingertech TinyESCs and their mixer. The weapon is powered by a 1,380 kv brushless motor and a Plush12 ESC from HobbyKing. The robot is powered

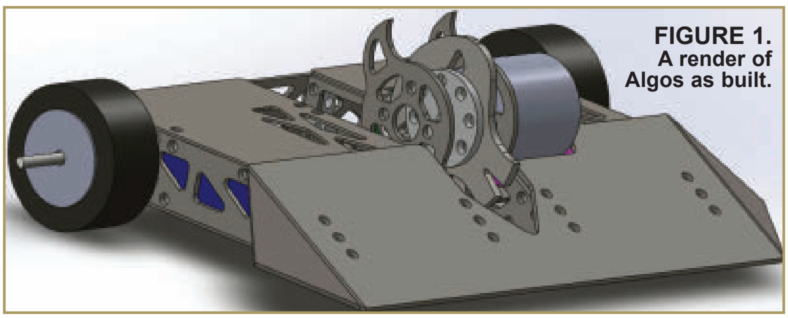


FIGURE 1. A render of Algos as built.

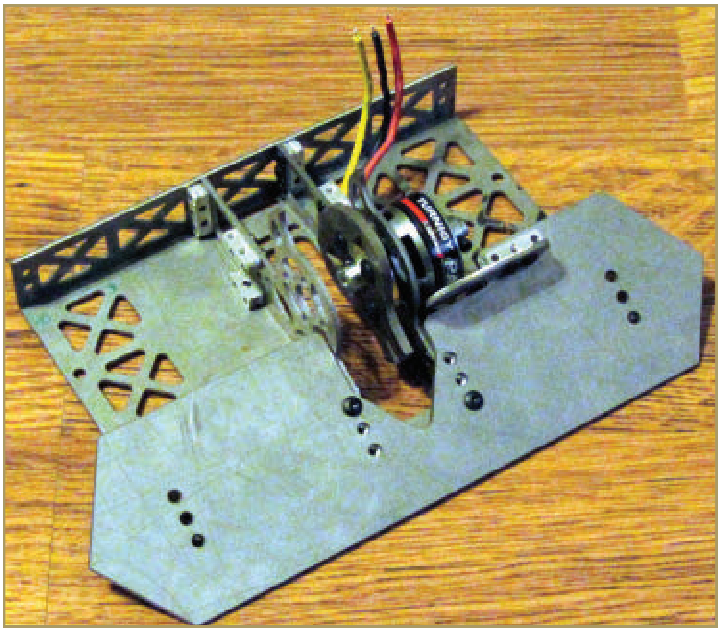


FIGURE 2. Central components of Algos installed to show the general real world layout.

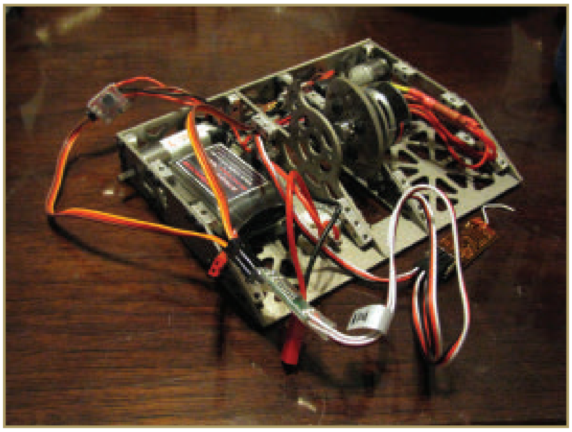


FIGURE 3. Test wiring of Algos to check components and program failsafe behavior.

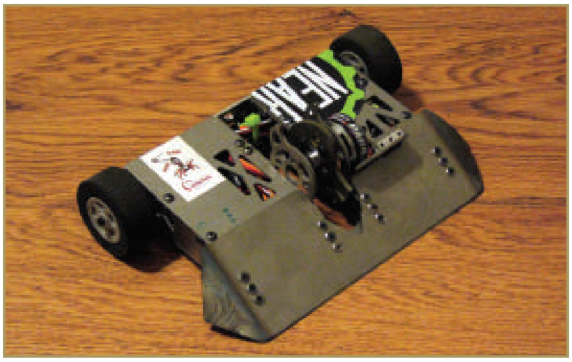


FIGURE 4. Algos prior to attending Clash of the Bots 3.

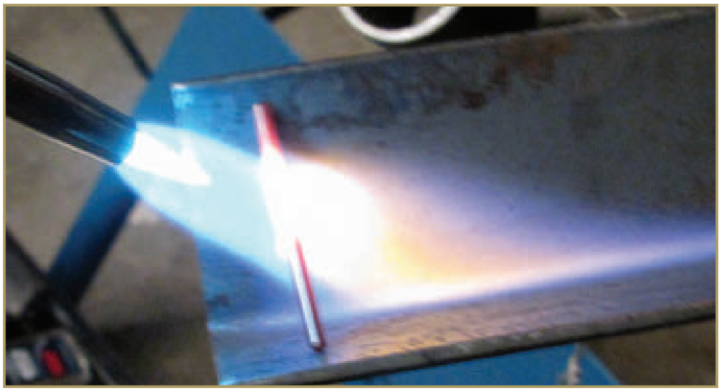


FIGURE 5. One of the shafts being heated on the improvised hardening rig.

by a 3s 325 mAh Thunderpower LiPoly battery and uses a custom power switch for easy startup and shutdown.

With this design, all of the chassis components are waterjet cut and held together with nutstrip from **kitbots.com**. This design method allows for rapid assembly and minimizes build time.

The electrical layout for Algos had to be adjusted during the build process. Initially, the plan was to have all of the components (besides the weapon ESC and one drive speed controller) tucked into the half of the chassis that was not occupied by the weapon motor.

During the build process, it became clear that due to the length of wires there would not be enough space for all of the components in the intended locations. To fix this, the receiver was moved to the opposite side of the chassis and the wires were rerouted to accommodate the new layout.

Once the electrical system was functional, the biggest remaining task was putting the bent corners in the 1/16" titanium front wedge. This was accomplished with the use of an oxy-acetylene torch, pliers, and several heavy blocks of aluminum clamped to a steel table.

The titanium was heated to orange hot along the bend line, and once an even color was reached I clamped onto the side and bent the edges down.

This method was not terribly accurate, but it was able to get the desired results. Once this was completed, I sharpened the wedge using an angle grinder, Dremel tool, and a file. At this point, Algos was ready for its debut event: Clash of the Bots 3 in Gastonia, NC.

Algos went 2-2 at Clash of the Bots and spent most of the time without the primary weapon. The stock shaft on the brushless motor had a groove cut in it for a retaining ring. This ring acted as a major stress concentration point and the result was the shaft snapping on every decent hit. With the next event less than two months away, a good and quick fix was needed.

I ordered a length of 3 mm diameter O1 tool steel from

FIGURE 6. **Same shaft material before heating and after quenching in oil (after being struck with a large hammer).**

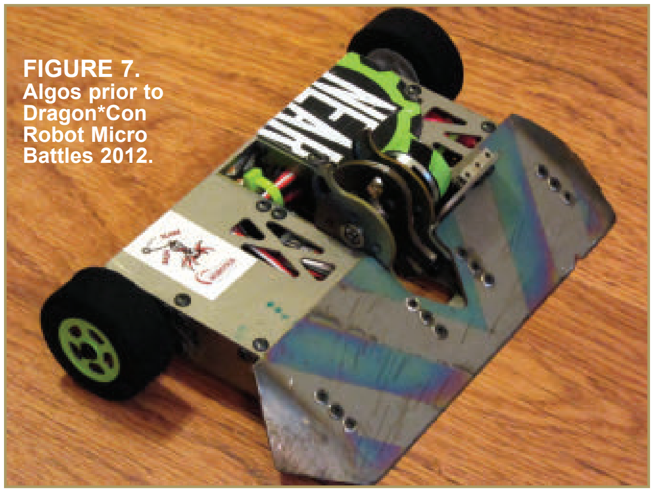


FIGURE 7. **Algos prior to Dragon*Con Robot Micro Battles 2012.**

McMaster.com and cut off several three inch lengths. These pieces were then put on my lathe and sanded down until they were able to slide into a 3 mm ball bearing I had purchased to add a second support to the shaft.

Once the shafts were sanded to the proper size, I tossed together a simple heat treating rig using a large chunk of steel angle, a steel can filled with 30 wt oil, and the same torch I used to bend the wedge.

The shafts were heated until they achieved a bright orange color and then were dropped lengthwise into the can of oil. Once cooled, the shafts were removed and cleaned.

At this point, the shafts were highly brittle and needed to be tempered in an oven.

The shafts were placed in a small aluminum foil cup and put in an oven for 45 minutes at approximately 450° F to properly temper the material. The tempering takes the steel back from a full hard to a usable hardness that will allow it to absorb impacts without shattering.

The final step in the rework was to remake the center rails since they had been damaged at Clash of the Bots 3 and they would need to be slightly adjusted to accommodate the extra bearing. While updating the design, I used it as a chance to add some material back into the design to bring Algos right up to the weight limit.

As a final touch, I decided to attempt to anodize the front wedge and center rails using an array of 9V batteries and Coke Zero. The process seemed to work well and the low current provided by the 9V batteries seemed to cause a slow transition through the color gradient that allowed a great deal of cosmetic control.

Algos placed 3rd at Dragon*Con Robot Micro Battles and sustained no damage. The shaft modifications performed perfectly, and Algos is ready for the next event. **SV**

EVENTS

Upcoming Event

Fall Fling will be presented by PennBots in Boiling Springs, PA on November 17th; **www.pennbots.org** **SV**

COMBOTs Cup Channel Now Up On YouTube

*COMBOTs —the annual robot fighting competition (**www.ComBots.net**) — now has their own video channel which features episodes of this popular event. Commentator Stephen Felk brings his own brand of quirky — yet insightful — narrative to well edited video highlights.*

*The episodes are available at **www.youtube.com/ComBotsCup**.*

BotBlast Turns Five

● by Dave Graham

Top fighting robot enthusiasts converged on Columbia Mall in Bloomsburg, PA, on Saturday, July 21, 2012 to celebrate the fifth anniversary of BotBlast (**Figure 1**). Thirty-six 150 gram Flea (a.k.a., Fairy), one pound Ant, three pound Beetle, and six pound Mantis bots slugged it out in the day-long competition for BotBlast 2012 bragging rights.

The 2011 BotBlast champions from all four weight classes returned eager to defend their titles. Event organizer Jeremy Campbell and Team Dreadfully Wicked Robots took a moment during the driver's meeting to recognize the five teams that have attended all five BotBlast events (**Figure 2**).

After the pleasantries, competitors turned their attention to business in the box as the drama and destruction of BotBlast 2012 began to unfold.

The action got off to a fast start in the opening Fleaweight match between two bots with spinning blades: my BotBlast 2011 returning champion Hedgehog and Giga Hurtz. I'd like to report Hedgehog dominated the match, but the truth is Giga Hurtz went into the self-destruct mode and Hedgehog came in to finish the job (**Figure 3**). Hedgehog went on to lose its next two matches and was eliminated.

The Fleaweight title came down to three contenders: Heath Hill driving BotBlast 2011 second place finisher wedge bot Baby V; Chris Atwood and his Motorama 2012 champion wedge bot Tracked Terror; and John Parsons and his beater bot Lolcat. Tracked Terror beat Baby V in the loser's bracket final to earn a title match with undefeated Lolcat.

The final match was an action-packed thriller between two deserving combatants that went the distance. It would be the first of many close calls for the judges this day, and was reflective of the quality of all the BotBlast competitors and their machines.

The judges awarded the decision to Tracked Terror, forcing a tie-breaker match in the tournament's double-elimination format. Lolcat dominated the tie-breaker chasing and flipping Tracked Terror the entire match. This time, the judges awarded Lolcat (**Figure 4**) the victory and the 2012 Fleaweight title.

The Ant competition got off to a disappointing start for me as I drew Brandon Nichols' wedge bot Bob Saget in the opening match. Prior to the competition, I had honed the tips of the spinning blade on my bot Kyle's Cutter. During the opening match, Bob Saget got under Kyle's Cutter forcing it into the wall. The blade stuck in the arena wooden bumper, forcing me to use my one "arena unstick."

Needless to say, I quickly found myself stuck in the arena wall a second time (**Figure 5**) and was counted out. Kyle's Cutter would return with its blade tips duct-taped to set a BotBlast knockout record with a 2.7 second win over Amatol, and then go on to dominate the Antweight rumble.

In Ant winner's bracket action, Sean McKeown and his wicked

FIGURE 1. Group shot of competitors at Columbia Mall.


FIGURE 2. BotBlast five year attendees.

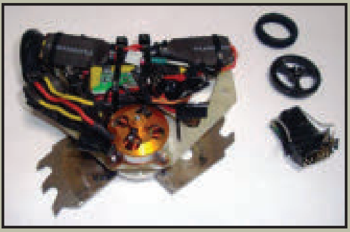


FIGURE 3. Fleaweight bot Giga Hurtz after match with Hedgehog.

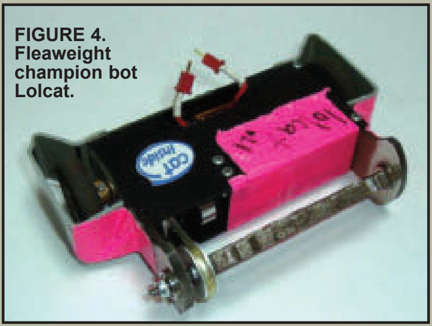


FIGURE 4. Fleaweight champion bot Lolcat.

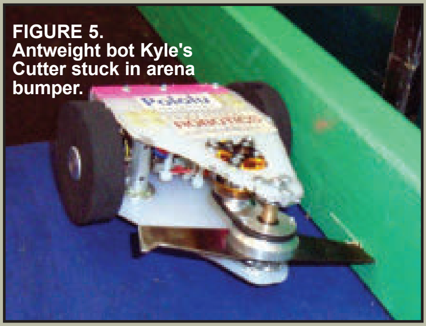


FIGURE 5. Antweight bot Kyle's Cutter stuck in arena bumper.

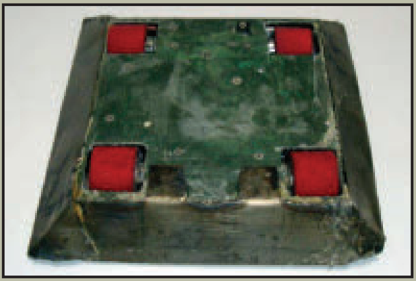


FIGURE 6. Antweight champion bot Little Box.

beater bot Catastrophic first sent Chris Atwood and his BotBlast 2011 champion wedge bot Antelope to the loser's bracket and then sent Cody Bennett and his spinning blade bot Szalor south. Catastrophic joined Antelope and Szalor in the bottom bracket after losing the winner's bracket final match to Richard Kelley and his titanium clad wedge bot, the Little Box (**Figure 6**).

In the loser's bracket, Szalor viciously ended Antelope's run at a repeat title by using its blade to skin the Antelope (**Figure 7**). Catastrophic then brutalized Szalor by literally ripping the front end off of the bot (**Figure 8**).

The title match was a rematch of the winner's bracket final, and the Little Box was again able to stay under Catastrophic's big spinning beater while repeatedly pushing Catastrophic into the wall to win the Antweight gold. Kelley also received the BotBlast Sportsman award.

The Beetleweight class was really a showdown between two big spinners: Gene Burbeck and his BotBlast 2011 champion One Fierce Lawn Boy, and Kyle Singer and his BotBlast 2011 third place finisher Ripto. On his way to running the winner's bracket, Ripto gave John Parsons and his undercutter Coercion a lesson in blunt force trauma (**Figure 9**), and then took a bite out of One Fierce Lawn Boy's titanium wheel guard (**Figure 10**).

Parsons would straighten Coercion's blade and come back to win the Beetleweight rumble. The final match between winner bracket Ripto and loser bracket One Fierce Lawn Boy was arguably the best Beetleweight match I have ever seen. The first half of the match was a slugfest with both bots delivering wicked hits. Midway through the match, Ripto lost its weapon drive belt and it looked like a sure win for One Fierce Lawn Boy. However, Kyle Singer's aggressive driving repeatedly forced One Fierce Lawn Boy into the wall, and as a result Lawn Boy was unable to deliver a killing blow to Ripto. The judges awarded Singer and Ripto a unanimous decision and the hard fought Beetle title. Ripto was also voted "Most Destructive" by its fellow competitors.

During the match, the lobe of Ripto's spinner was broken off by One Fierce Lawn Boy. That really surprised Singer who attributed it to a stress fracture that occurred during the spinner hardening process. The stress fracture is clearly visible (the dark rectangle on the lefthand side of the spinner) in the close-up view of the spinner (**Figure 11**).

The Mantis competition showcased five bots, including Zac O'Donnell's flipper bot Threecoil (**Figure 12**), winner of the "Coolest

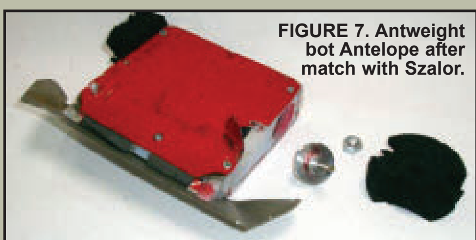


FIGURE 7. Antweight bot Antelope after match with Szalor.

FIGURE 8. Antweight bot Szalor after match with Catastrophic.

FIGURE 9. Beetleweight bot Coercion after match with Ripto.

FIGURE 10. Damage to Beetleweight bot One Fierce Lawn Boy after match with Ripto.

FIGURE 11. Stress fracture on Ripto's spinner (dark area on the left).

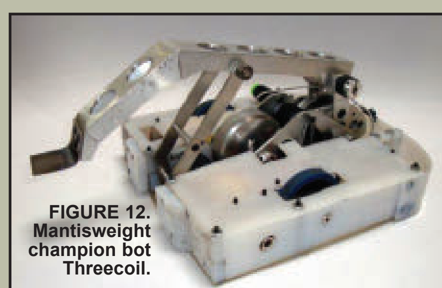


FIGURE 12. Mantisweight champion bot Threecoil.

FIGURE 13. Mantisweight bot One Fierce Bush Wacker.

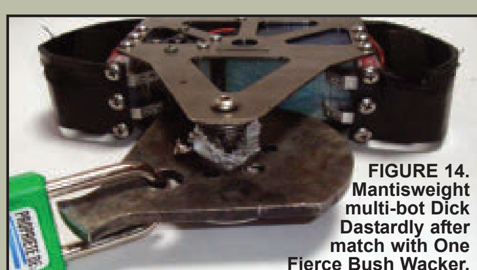


FIGURE 14. Mantisweight multi-bot Dick Dastardly after match with One Fierce Bush Wacker.

FIGURE 15. Close-up of damage to Mantisweight multi-bot Dick Dastardly's spinner.

Robot" and "Best Engineered" awards; a multi-bot; and Gene Burbeck's returning 2011 champion, the undefeated wrecking machine called One Fierce Bush Wacker (**Figure 13**).

After two rounds of round-robin competition, both Threecoil and One Fierce Bush Wacker were undefeated, and Dead Metal and multi-bot Muttley and Dick Dastardly were winless. In the opening match of round three, Chris Atwood drove Dead Metal to a victory over the multi-bot securing a third place finish. Chris also received the "Best Driver" award.

The championship match pitted the raw spinning power of One Fierce Bush Wacker against the complex flipper of Threecoil. One Fierce Bush Wacker had already demonstrated that destructive power on the multi-bots by ripping the wheels off of Muttley and then seriously damaging the weapon assembly of Dick Dastardly (**Figures 14** and **15**).

After losing badly to Burbeck and the Bush Wacker last year at BotBlast with an innovative but ineffective flipper bot named Reclipso that simply failed to launch anything, O'Donnell again found himself facing Burbeck and the Bush Wacker in the final match. This time, it was his redesigned and renamed flipper bot Threecoil that would answer the bell.

Changes to O'Donnell's bot included the addition of a flywheel to the flipper mechanism making it an effective weapon that could repeatedly fire. But could Threecoil and its complex flipper mechanism survive repeated massive hits from Burbeck's proven killing machine?

The match started with the Bush Wacker heading straight for Threecoil, but the big spinning blade was deflected by Threecoil's armor. Then, Threecoil lifted the Bush Wacker and moved it — didn't flip it, just lifted it and pushed it back a bit. It was a scene that would be repeated throughout the match — the Bush Wacker unable to get a bite on Threecoil, and Threecoil lifting the Bush Wacker pretty much at will.

About halfway through the match, it appeared the Bush Wacker lost one of its drive wheels making Threecoil's job easier. You could sense Gene Burbeck's frustration toward the end of the match when the whine of the Bush Wacker's blade increased as Gene throttled it up.

The match went the distance and the judges awarded a unanimous, well-deserved decision to winner Zac O'Donnell and his engineering marvel — the 2012 BotBlast Mantis champion, Threecoil.

A list of the winners is shown in **Table 1**, and a list of the special award winners as either voted on by the competitors or selected by the event organizer is shown in **Table 2**. The winners shared a bounty of prizes, including custom designed sequenced starting lights for first place, trophies and plaques for second place, third place, rumble winners, and special awards, and tools and robot building components donated by sponsors.

Event organizer Jeremy Campbell wants to thank his five year sponsors the Columbia Mall, Sears, and Mainville Garage; his returning sponsor Fingertech Robotics; and his new sponsors Pololu and *SERVO Magazine*. Campbell also credits his mother Trish with managing registration and the pit area, and father Warren as the best arena man in the sport, taking care of virtually every aspect of the arena down to the custom artwork on the arena floor.

I just can't say enough good things about BotBlast. It's a well organized and well run event in a great venue with the best bounty of prizes you'll find anywhere. Next year promises to be even better as Campbell and his family have promised a new 12 x 12 foot steel arena (no more wooden bumpers, thank goodness!), and the addition of at least one heavier weight class.

Mark your calendar now for mid-July 2013, and plan to attend this fun event! You can follow BotBlast on their website at **www.botblast.webs.com**. SV

TABLE 1. BOTBLAST WINNERS.

	Fleaweight	Antweight	Beetleweight	Mantis
1st:	Lolcat *John Parsons*	Little Box *Richard Kelley*	Ripto *Kyle Singer*	Three Coil *Zac O'Donnell*
2nd:	Tracked Terror *Chris Atwood*	Gyroscopic *Sean McKeown*	One Fierce Lawn Boy *Gene Burbeck*	One Fierce Bush Wacker *Gene Burbeck*
3rd:	Baby V *Heath Hill*	Szalor *Cody Bennett*	D-12 *Brandon Nichols*	Dead Metal *Chris Atwood*
Rumble:	Baby Box *Richard Kelley*	Kyle's Cutter *Dave Graham*	Coercion *John Parsons*	Dead Metal *Chris Atwood*

TABLE 2. SPECIAL AWARD WINNERS.

Longest Distance Traveled:	Gene Burbeck and fiancée Cathi (Michigan)
Sportsmanship:	Richard Kelley
Best Driver:	Chris Atwood
Best Engineered:	Threecoil
Coolest:	Threecoil
Most Destructive:	Ripto

PARTS IS PARTS:

REVIEW: MicroLux Miter/Cut-Off Saw

● by Pete Smith

The small cut-off saw in **Figure 1** is distributed by Micro-Mark **www.micromark.com**) and has proved useful when cutting aluminum nutstrip to length. The saw comes with a wood/plastics blade as standard, but abrasive disks are available in packets of three for an additional cost. (I used the abrasive disks when cutting the nutstrip.)


FIGURE 1.

The instructions for replacing the disks are not totally clear, and a little perseverance was required to get the abrasive disk installed.

The blade has an effective safety guard, but safety glasses are still a necessity with this sort of saw.The saw has a vise that can be adjusted from 0-45º, and it has a groove to allow it to grip round stock (**Figure 2**)

I used it to cut micro, mini, and regular 3/8" x 3/8", and it managed all three (maximum capacity is 1/2 x 1/2). You need to let the disk do the work because forcing it will stall the motor. It produces neat cuts that require minimal filing. The saw does produce a considerable amount of fine aluminum filings, so don't use it near electronic devices.

It's not really faster than using a hacksaw and file, but that gets tedious very quickly when cutting a large number of parts. The saw definitely produces a neater cut that requires less cleaning up. SV

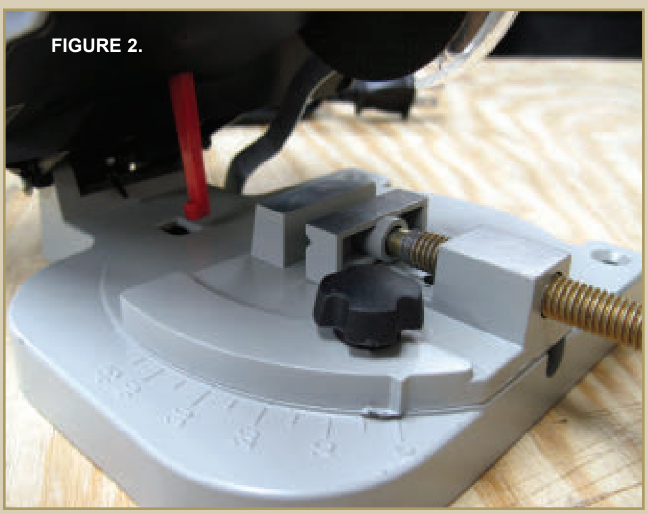


FIGURE 2.

Build the Kronos Flyer

by Michael Simpson

Part 1: Introduction into Multi-Rotors

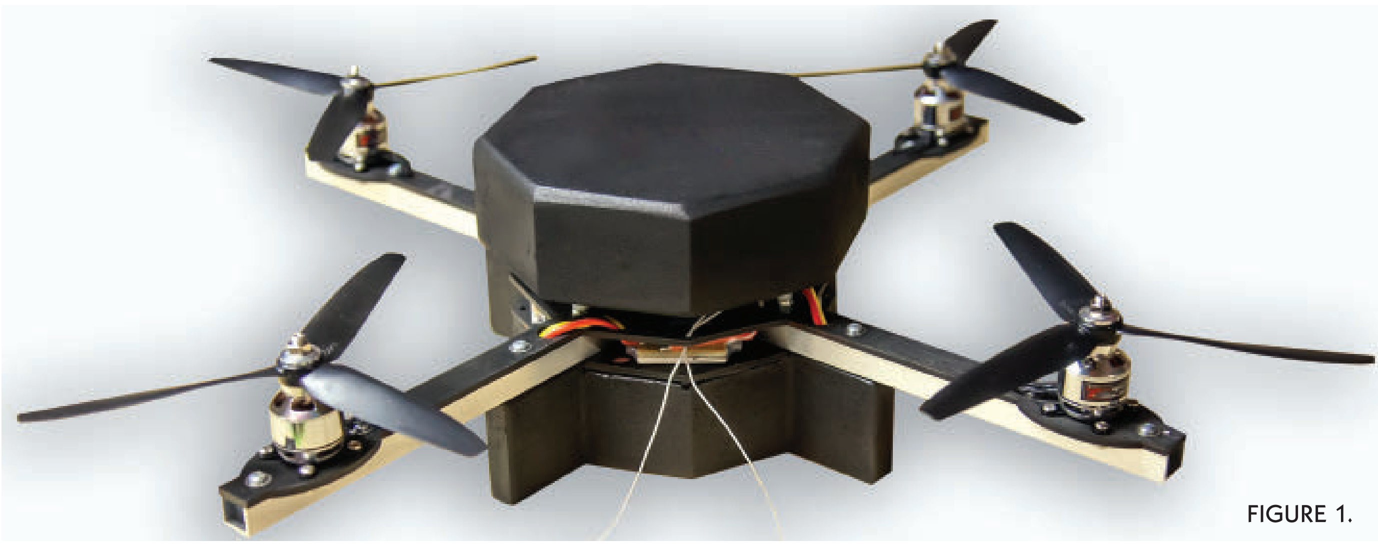


FIGURE 1.

www.servomagazine.com/index.php?/magazine/article/november2012_Simpson

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

I started playing with R/C helicopters a while back, and was always interested in automating them in one way or another. However, it wasn't until multi-rotor craft started to become prevalent that I really took a serious interest.

One of the very first articles I wrote for *Nuts & Volts* was back in 2002. It was about my Kronos Crawler. This time, we're going to build the Kronos Flyer. The Kronos Flyer is a quadcopter. That is to say it has four stationary motors that drive four fixed pitch props. The onboard computer controls the speed of the props, thus allowing you to control the yaw, pitch, and roll of the copter electronically.

Whether your desire is to build an automated craft or to fly an FPV (First Person View) craft, you have to crawl before you can walk, run, or fly. It is important for you to be able to take control of the craft in an emergency.

In this five part series, I will show you how to build and tune a quadcopter that can take a pounding. Why does it need to take a pounding? Because you are going to crash. A lot. Some crashes will be minor, others will be horrific. Flying a quad is not easy.

Why Build Your Own?

If you make the booms and platform parts yourself, the Kronos Flyer will cost you about $200 to build. I urge you to do this because you will end up with extras that you can use to build another quad or replace broken parts.

For instance, the aluminum C-channel comes in 96" lengths from most home centers. It's priced around $11 and will yield 11 booms. You only need four to start, so that makes for a lot of replacements.

Multi-Rotor Types and Sizes

Multi-rotors are (for the most part) broken down by size and type. Sizes are measured in millimeters from the motor centers. There really aren't any special rules when it comes to size, and there is a lot of overlap.

Size Groups

- Nano: Less than 100 mm.
- Micro or Mini: 100 mm - 300 mm.
- Mid: 300 mm - 500 mm.
- Large: Greater than 500 mm.

Types of Multi-rotors

- Tricopter: Three blades and one tail servo.
- Quadcopter: Four blades in a +, X, or H configuration.
- Hex: Six blades in a +, X, or Y configuration.
- Octoquad: Eight blades in various configurations.

Why not make the Kronos Flyer a tricopter or a hexcopter? Well, a quadcopter is a little more stable than a tricopter and has more lift potential. The tricopter is a more complicated build. A hexcopter is actually more stable and has more lift potential than a quadcopter, but the problem is that the hexcopter has more parts and is therefore harder to build and costs more. A quad gives you the best overall bang for your buck. As for size, the smaller the craft, the less damage it sustains when you crash. Plus, you can usually fly small craft indoors. The problem with small craft is that they have very little lift potential, but this just means they make great general-purpose trainers.

FIGURE 2.

Larger multi-rotor craft are very stable and can lift more payload. They also handle the wind much better than their smaller counterparts. The problem with larger craft is that you need more space, money, and time to fly them. They can also be quite dangerous. The size of the Kronos Flyer (**Figure 2**) is 17" from motor center to motor center, or 430 mm. This build seems to be the perfect size for cost, complexity, and maintenance. Its aluminum construction makes it durable. I would not fly it around indoors, but you may be able to practice hovering it in a large basement or garage.

Multi-Rotor Parts — the 1,000 Foot View

Here's a brief overview of all the components that make up a multi-rotor. This will help later when we order parts and assemble our Kronos Flyer. This is by no means the end-all in parts and accessories. I will touch only lightly on each component.

FIGURE 3.

Frames

The frame component — like the ones shown in **Figure 3** — can be purchased or scratch-built. The frame is made up of a series of booms that connect to some sort of center platform. The ends of the booms have a means to mount to a motor.

Frames can be made from aluminum, plastic, fiberglass, carbon fiber, or wood. Most likely they are made from a combination of these materials. Each material type has its advantages and disadvantages.

Motors

Most multi-rotors use brushless motors. A few years back, brushless motors were a rarity in R/C because they were outrageously expensive. Now they are the norm, and the prices have come down.

They run smoother and quieter than their counterparts. They have less wear since you don't have rubbing parts. They also produce less electrical noise.

Brushless motors come in inrunner and outrunner varieties. The inrunner is much like your traditional brushed motor with the shaft rotating inside a housing. The outrunner variety has the rotating portion of the motor on the outside.

Outrunner brushless motors tend

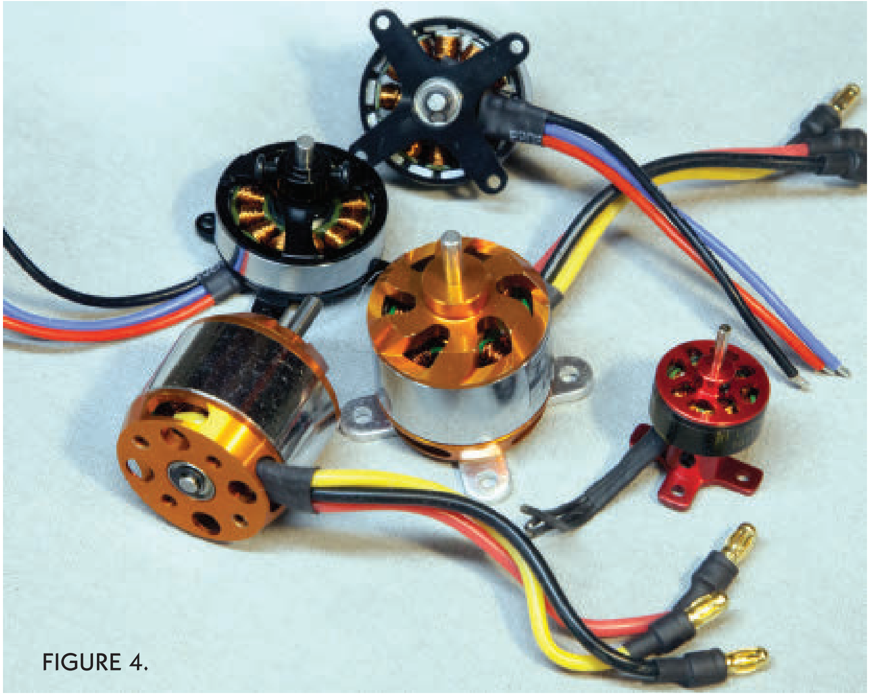


FIGURE 4.

to spin slower and have more power than inrunner motors. Most multi-copters are built using outrunners like the ones shown in **Figure 4**.

Brushless motors tend to be labeled in three different ways.

1. Can Size. This is based on a system that Mabuchi Motor Company came up with a while back for brushed electric motors for R/C cars. It has now made its way into the brushless electric motors for cars. Since some builders use these same motors on R/C planes and helicopters, you will see them used for multi-copters, as well.

Here are a few of the can sizes:

- 370 = 27 mm x 34 mm
- 380 = 27 mm x 38 mm
- 550 = 37 mm x 57 mm
- 700 = 40 mm x 71 mm

2. Glow Engine Size. In this case, the brushless motors are rated at what an equivalent R/C gas glow plug of a given displacement would be. Since brushless motors don't really have a displacement, it's only a

FIGURE 6.

Propellers

Propellers (like the ones shown in **Figure 6**) come in many shapes and sizes. Unlike R/C motors, R/C props have a rating system that makes them very easy to compare. An 8x4 prop will have a diameter of 8" and a pitch of 4". The pitch of a prop is the theoretical distance in inches the prop will move forward in one revolution. A pitch of four means in a perfect world, the prop would move forward 4" for each revolution.

A prop with an 8x6 rating will give us much more thrust than a prop with a rating of 8x4. The 8x4 prop will also cause our motor to draw more amps. Smaller props are used on faster motors, while larger props are used on motors that run slower. In addition to the size and pitch, a prop can also be designed to run in a counter rotation. These props will be designated with an R added to the rating number. Sometimes a P (for pusher) is used. For a quadcopter, you need two normal props and two counter-rotating props. This is done to counteract the propeller torque effect.

FIGURE 7.

Props have to be balanced. This is something you must do to every prop you place on your multi-rotor. In order to balance a prop, you need a prop balancer like the one in **Figure 7**.

comparison of power. For example, a .15 brushless motor will have the same power as a gas engine with a .15 displacement. Also be aware that many times a .15 gas or electric motor will be referred to as a 15.

3. Stator Size. On an outrunner brushless motor, the rotor is the outer part of the motor. It has permanent magnets attached to it. The rotor is the part that turns. This is what you connect your propeller to. The rotor is wrapped around the stator. The stator contains the coils which — when energized in a particular order — cause the rotor to turn. Many manufacturers now use this when describing their motors. For instance, a 2826 motor will have a stator that is 28 mm in diameter and 26 mm long. Along with the stator size, a kv rating is given. The kv rating is the rpm per volt of the motor. So, a motor with a rating of 2,826-2,900 kv will turn 21,460 rpm with 7.4 volts.

With these ratings, you can start to compare motor sizes. Even motors that are rated using the can or glow engine size will have kv ratings. Some will also give you the stator size, as well.

Some manufacturers will provide the rotor size and not the stator size. Take the Turnigy 2211 motor shown in **Figure 5**, for example. The stator is actually 1809. It's not that big of a deal as long as you know that they are doing this.

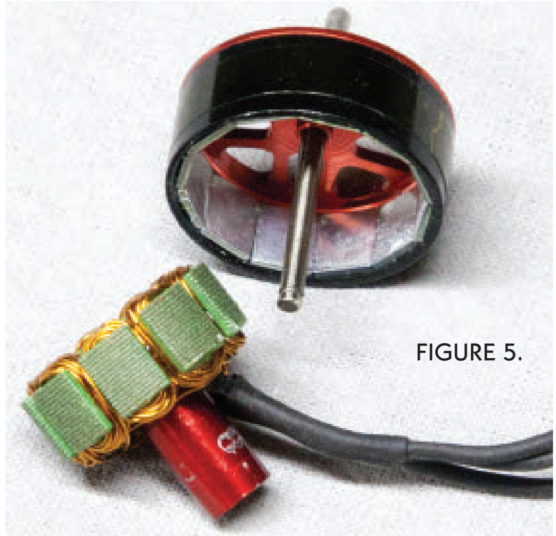


FIGURE 5.

A lower kv rating means lower speed and more torque. These are good for driving large props. A higher kv rating means less torque and more speed. These are better at driving small props. A motor will also have a cell rating and max amperage rating. If you exceed either of these, you could burn up the motor. The amp rating will give you an idea of how much power the motor will draw. If a motor draws 40 amps and you have four of them running, you are going to drain your battery quickly unless you have a very large battery.

ESCs

While there are many Electronic Speed Controllers available, it is very easy to choose one. First, since we are using brushless motors, we need a brushless ESC. You choose an ESC based on its amperage rating. Two ESCs are shown in **Figure 8**. The larger one is 30 amps; the smaller one is 12 amps.

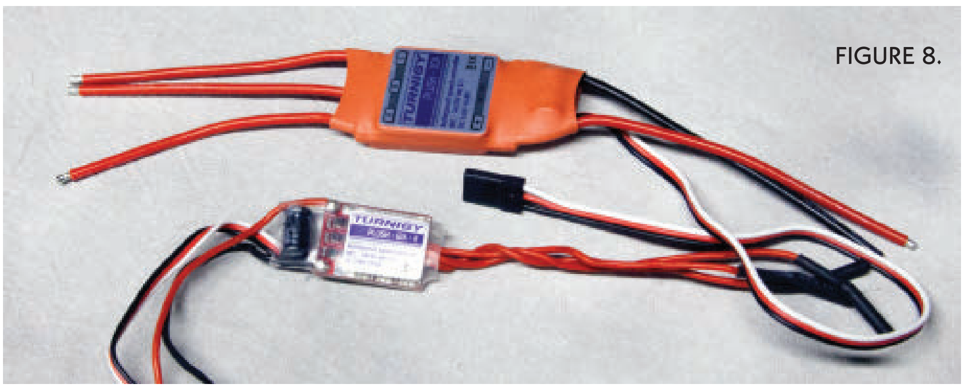


FIGURE 8.

If your motor's maximum amperage is 25 amps, you will need the 30 amp ESC. If you have a small craft that only pulls 10 amps, then the 30 amp ESC would be overkill and add unnecessary weight to the craft. In that case, you would choose the 12 amp ESC.

Various ESCs have features that you may want. For instance, on multi-rotors we use small computers for control. We want an ESC that is capable of responding quick enough to keep up with the controller. The type of connectors that come with the ESC may dictate which one you choose for a given project.

Batteries

In order for your quadcopter to get off the ground, it needs power. We will be using Li-Po batteries as our power source. A Li-Po's power-to-weight ratio is the best choice at this point in time. They have more capacity than their predecessors, so for the same storage capacity you have much less weight.

Li-Po batteries (like the ones shown in **Figure 9**) come in many different sizes, shapes, and configurations. They use a rating system that shows the number of cells in the pack. A battery marked 3S1P means that it is three cells connected in series to make the battery. A 2S1P will have two cells connected in series.

The last two digits are how may parallel packs there are. For instance, if it's a 2,200 mAh 3S1P, it has three 2,200 mAh cells connected in series. If it's a 4,400 3S2P, then it has two 2,200 mAh cells connected in parallel, then three of those connected in series. If it is only a 1P battery, the manufacturer will often leave this part of the nomenclature off.

Li-Po batteries also have discharge ratings. This will be marked as an *XX*C number. A 25C means that it can be discharged at a rate of 25 times its capacity. So, in the case of a 2,200 mAh battery, you could discharge it at a rate of 55,000 mAh or 55 amps, without the battery being damaged.

Speaking of damage, Li-Po

batteries are very susceptible to permanent damage. Do any of the following and you might as well take your battery to the recycling center:

- Drain the battery below three volts per cell.
- Overcharge the battery.
- Charge the battery too fast.
- Discharge the battery too fast.
- Fail to balance the battery.
- Apply a reverse voltage to the battery or cell within the battery.

You need a special charger to charge Li-Po batteries. The one shown in **Figure 10** is very inexpensive, and can discharge and balance your batteries.

What Exactly is Balancing?

It is very important that each cell in a Li-Po pack be at the same voltage. If they aren't, draining a multi-cell battery could result in one of the cells dropping too far and getting damaged. Each multi-cell Li-Po comes with a balancing connector in addition to its normal connector. The balancing connector is actually a tap into each cell in the pack. This allows the charger to look at the individual voltages of each cell.

During a balance charge, the battery is charged and then the higher voltage cells are each discharged until they reach the lowest cell in the pack. You don't have to do a balance charge each time you charge, but it should be done every couple charges.

If you treat them well, Li-Po batteries will last a long time.

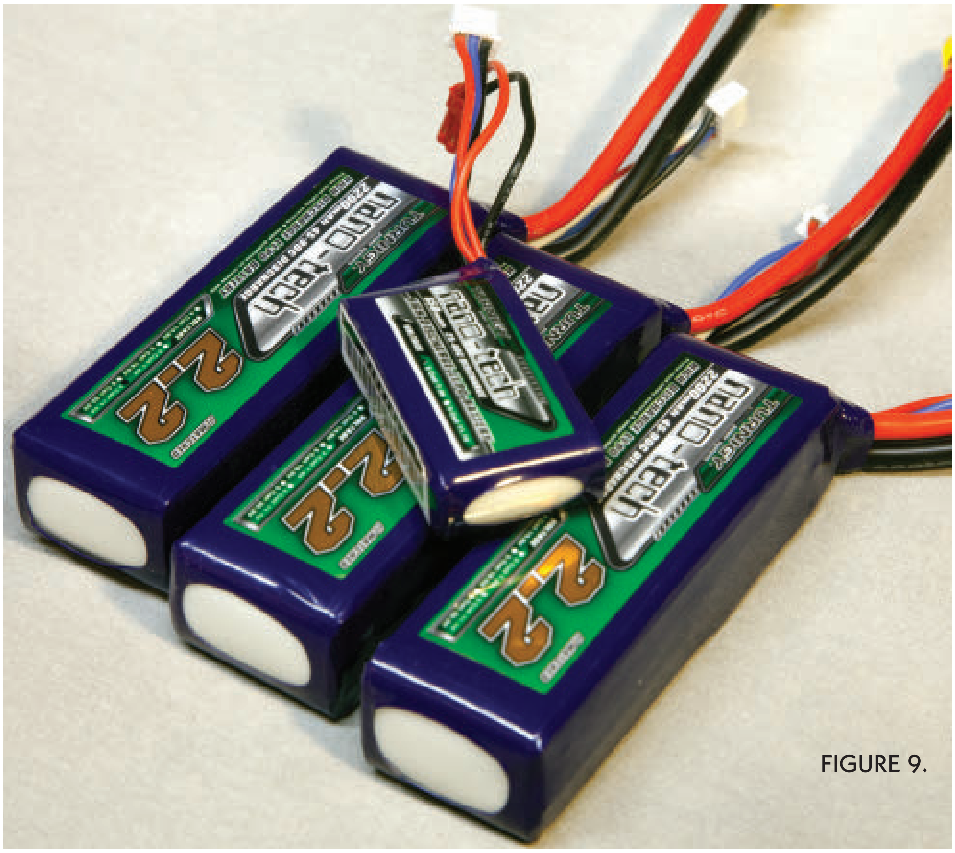


FIGURE 9.

FIGURE 10.

The Radio

To run a quad, you can do it with a four-channel radio, but this leaves no room for expansion. For this reason, the minimum I recommend for controlling a quad is six channels.

All the radios shown in **Figure 11** are 2.4 GHz radios except for the one in the upper left corner; it is an older 72 MHz Futaba six-channel helicopter radio.

Hobby King HK6S

This radio is the least expensive radio I own. I have used it to fly multiple quads and to control various robots. It feels cheap, because it is. The auto centering controls never truly return back to center, so it can be a little difficult to fine-tune a quad. I don't think I would trust an $800 quad to this radio.

The transmitter uses four AA batteries. The receiver seems to be better quality than the transmitter.

Tactic TTX404/TTX600

I purchased two of these radios from a local hobby shop. These were used to allow spectators to control two battling robots at the DC Science and Engineering Fair held

earlier this year.

They ran nonstop for three days, and still work perfectly. I might also add that it ran three days on the same set of batteries.

I also had a chance to use the Tactic TTX600. This is the six-channel version of the radio, and would be recommended if you want a low radio with a little expansion capability.

The only downside to these radios is that the throttle has detents — not good for helis or quads where precise throttle control is needed. I did open up my radio and file down the detents, so it works a lot better.

Futaba 8FG Super

I can't say enough about this radio. It's a standard eight-channel radio, however, if you use the S-bus function you get 12 proportional channels and two switched. Every single knob, switch, or trim is assignable.

There are eight switches — six of which are three-position switches. There are two knobs on the front and two sliders on the back. You have your two main joysticks that control four axes. There are four digital trims that can be assigned as you see fit.

The radio will store 20 models and has a slot for an SD card so that you can store more.

All of these features are nice, but the main reason I purchased the 8FG was that many of the new flight controllers coming out support the S-bus.

This means just a single wire from the receiver to the flight controller, not eight. (More on this when I talk about the flight controllers in the next section.)

Radio Cross-Compatibility

There is no cross-compatibility. A receiver designed for the Tactic will not work on the Futaba. They all have their own proprietary systems for managing the 2.4 GHz frequency spectrum.

There is one exception to this. Since the Futaba receivers are so expensive ($140 or more), many of the other manufacturers are making

FIGURE 11.

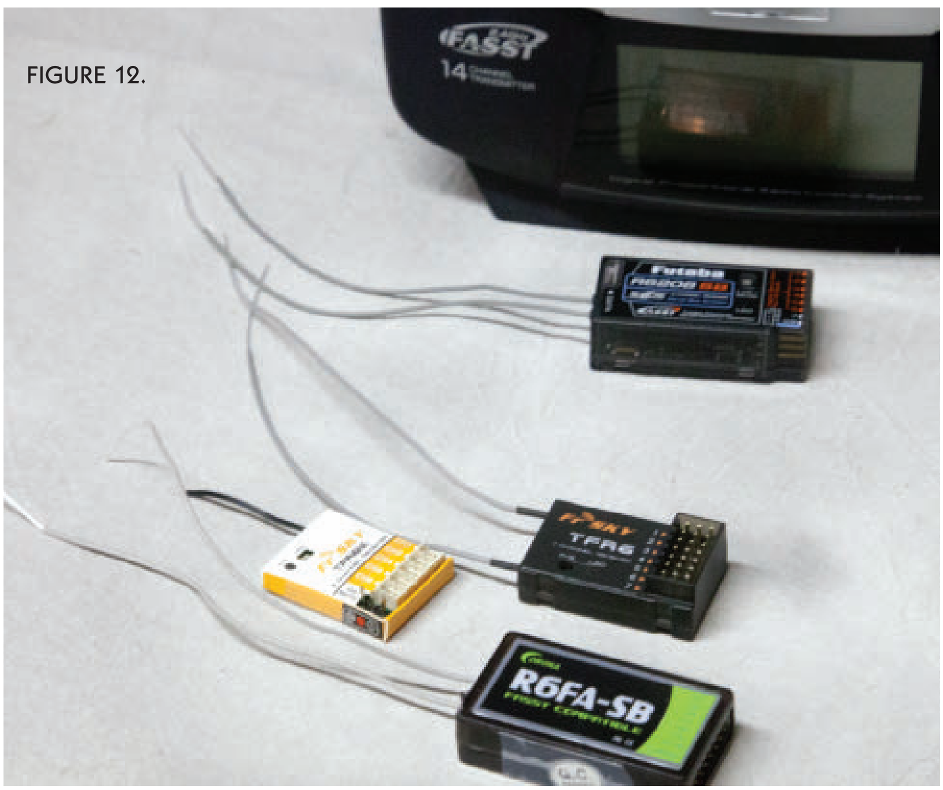


FIGURE 12.

FASST compatible receivers for use with the Futaba radio systems.

The three radios shown in the foreground of **Figure 12** are receivers from different manufacturers, and each works perfectly. The Corona R6FA-SB is even S-bus compatible.

In addition, Tactic sells a device called an Anylink. It plugs into the back of my Futaba and lets me transmit to all my Tactic receivers. I am able to use my Futaba 72 MHz radio on the 2.4 GHz band.

One last thing about selecting the radio to use with your multi-rotor. While simple radios like the HK6 and the Tactic are okay for the basics, you want a radio that gives you configurability, and I don't just mean reversing control for direction and trims.

A good example is when I went to test the Multiwii board — the only radio that would work was my Futaba 8FG.

I had to go into the settings and adjust the endpoints so the MultiWii would recognize the basic stick commands used for arming and calibrating the board.

You want to look for a radio that has (at the very least) the following configuration settings:

- Trims for each control.
- Reverse for each control.
- Ability to adjust the endpoints for each control.
- Multiple model storage for the parameters.

Look for a radio that has these features:

- 2.4 GHz.
- At least six channels.
- Rechargeable batteries for transmitter.

Throttle curves and sub trims are also a plus. If you select the right radio, it will serve all your R/C needs — be it flying a multi-rotor or controlling a ground based robot.

Controllers

The single most important component in a multi-rotor is the tiny computer called the flight controller (shown in **Figure 13**). The flight controller uses gyros and accelerometers to sense its current attitude.

It then controls the speed of each of the motors to provide any corrections to keep the craft level. At the same time, the controller is monitoring the input from the receiver so that it can factor in your transmitter joysticks.

For example, when you increase the throttle it will increase the rpm of all motors by the same amount. If you move the aileron control to the right, it will raise the rpm of the motors on the left and decrease the rpm of the motors on the right. It does all this while also taking into account the gyros and accelerometers.

Some of the flight controllers have additional sensors like barometric sensors to help with altitude holding. Still others have upgrade capabilities that add GPS and compass sensors. These will allow even more stabilization of your craft.

In this series, I will be touching on three particular controllers:

- Hobby King KK2
- DJI NAZA
- Multiwii V2.0

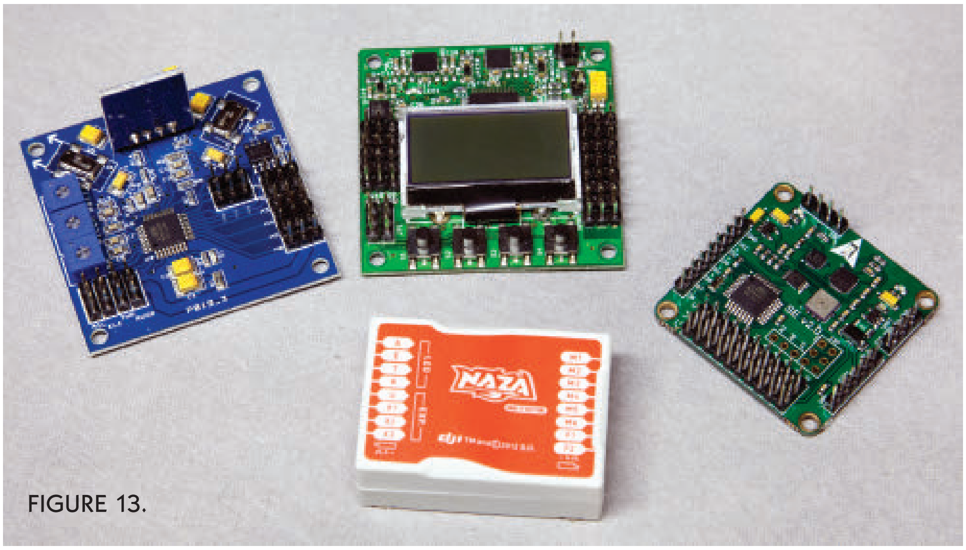


FIGURE 13.

I will cover mounting each of these in the Kronos Flyer. I will also cover the setup interface for each one. The actual settings and tuning procedures and detailed reviews for each controller will be posted on the Kronos Flyer web page.

In the fourth article of this series, I will include as much of this information as space permits. You will need to keep in mind that the settings and procedures for tuning will change as firmware upgrades are made available to the various controllers.

When sensors are added, variations in design — and even personal preferences — will affect the settings.

What's Next?

Next time, I will be concentrating on the actual design of the Kronos Flyer. I will provide you with a complete breakdown of all the parts. This will give you a chance to collect everything for assembly in the third installment. For now, you need to start thinking about the radio you are going to use to give your Kronos Flyer wings. SV

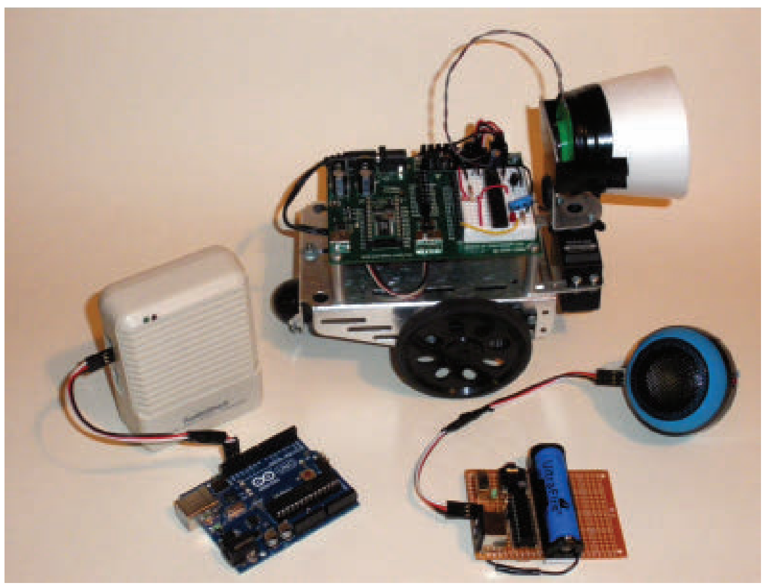
REVIEW

TALK IS CHEAP!

Low Cost Speech for Arduino, PICAXE, and BASIC Stamp Projects

by Eric Ostendorff

www.servomagazine.com/index.php?/magazine/article/november2012_Ostendorff
Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com



When *Star Wars* came out in 1977, the world loved R2-D2 even though he couldn't talk. He expressed himself with an amazing variety of chirps, beeps, and clicks. No doubt inspired to help computers and robots talk, American technology rose to the occasion. By the early 1980s, the speech synthesis market was booming. Hobbyists could buy General Instruments' SP0-256 speech synthesizer and companion TTS chip at RadioShack. The "Shack" and Heathkit used Votrax SC-01 based synthesizers for their TRS-80 computers and HERO robots, respectively. Even Target carried the "Voice Messenger" which gave the Commodore 64 decent text to speech (TTS) capability. Texas Instruments had speaking products for computers and toys. Who doesn't remember the Speak & Spell toys?

Surprisingly, stand-alone TTS chips nearly died off in subsequent years. The surplus of Votrax and General Instruments chips supported a trickle of hobbyist interest, but they got increasingly difficult to find this millennium. Ken Lemieux of **speechchips.com** was one of several suppliers of discontinued chips and hard-to-find parts — including the SP0-256. More recently, Ken carried the Speakjet, Soundgin, and Babblebot speech chips. Brand new as of July 2012 is the SP0-512, a.k.a., RoboVoice. Inspired by his longtime association with the SP0-256, RoboVoice is Ken's own creation using modern technology. He spent several months coding in ANSI C and created a one-chip synthesizer using 800+ TTS rules.

The 3.3V dsPIC chip has a built-in 16-bit, 16 kHz digital analog converter, so no external filter is required. Likewise, no external crystal is required like the earlier chips needed.

In fact, just a handful of parts (two capacitors, two LEDs, and five resistors) are required to make RoboVoice operational, plus a 3.3V supply, an audio amplifier, and speaker. Speech quality is robotic and very similar to the SP0-256 with its companion TTS chip. It's a blast from the past!

Ken currently sells just the RoboVoice chip, but a PCB (printed circuit board) is in the works. Users are free to experiment on breadboards or build it into their own design. Speech data is sent through a simple serial connection; 9600 baud true 8N1. Typically, three connections are required to interface to most microprocessors: serial in, speaking/busy, and ground.

In a pinch, you can get by with just two connections, without any speaking feedback. **Figure 1** shows my personal simplified version of Ken's schematic (which is available in the datasheet at **www.speechchips.com/downloads/SP0-512-Datasheet.pdf**).

Construction Details

My speech experiments used RoboVoice chips in four different configurations for use with three popular controllers: a BASIC Stamp 2, a PICAXE 20M2, and an Arduino Uno. Two RoboVoice chips were built into existing speaker/amplifier modules for use with nearly any processor. Another chip was breadboarded onto a Parallax BoeBot mobile robot. The fourth circuit is a simple keyboard terminal for use with one of the speaker/amplifier modules.

I didn't use RoboVoice's serial Tx (pin 4) since handshaking/flow control isn't implemented; the only data coming out of that pin is a welcome message with a version number. Since this 3.3V device is likely to be hooked up to a 5V device, you can use either the level shifter shown in Ken's schematic, or (as I did) series 10K resistors on both the serial Rx line (pin 6) and the busy/speaking line (pin 17).

The datasheet says the pins are "5V tolerant," but better safe than sorry. Resistors are cheaper than a new chip!

Small amplified speakers are all the rage now for plugging into MP3 players and phones as shown in **Figure 2**. They come in a wide variety of shapes, sizes, and colors and make great housings for the RoboVoice circuit after a little hacking.

RadioShack's $15 mini amplifier/speaker (**www.radioshack.com/product/index jsp?productId=2062620**) is LM386-based, powered by a 9V battery, and has a rotary volume control. I removed all three jacks from the PCB (audio in, audio out, DC in) to make room for a small piece of perfboard with the 512 and circuitry (**Figure 3**). Mine is a tight fit since I used a huge TO-220 case 3.3V regulator; a smaller

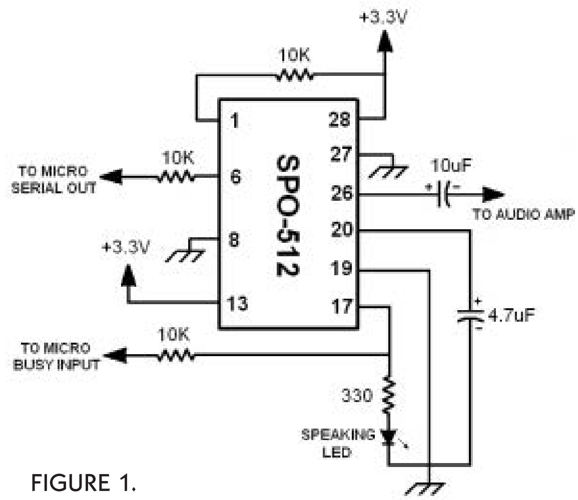


FIGURE 1.

ROBOVOICE SIMPLIFIED CIRCUIT


FIGURE 3.


FIGURE 2.

100 mA unit would work fine, as this is a low power application.

I drilled two holes on the front of the case to mount two 3 mm, 1 mA indicator LEDs (**junun.org**). A male three-pin header sticks out of the case where the audio jacks were originally, so I can use a servo style cable to connect to various controllers. Servo extension cables are cheap from eBay, and they can be cut and spliced to make the required connections.

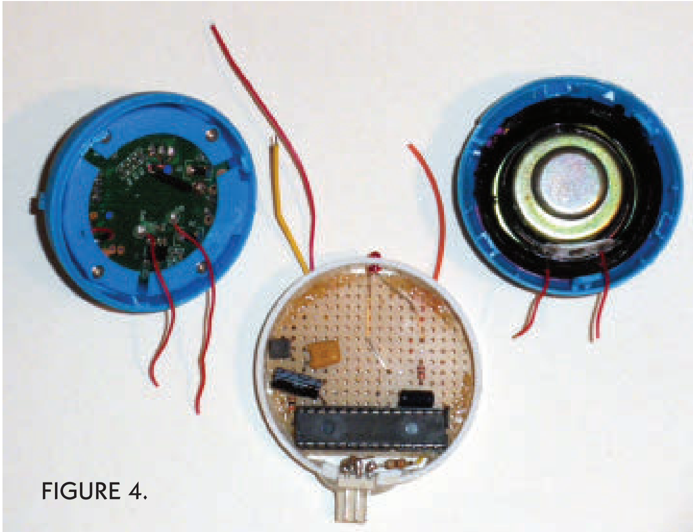


FIGURE 4.

Internal connection points for power and audio are easily traced with a multimeter; you ARE a hacker, after all! RoboVoice pin 26 goes through a 10 µF cap into the amplifier input.

Amplified "hamburger speakers" are under $5 from eBay China. They are small, loud, and have an internal lithium battery recharged via a USB cable. They are perfect for mounting on a small mobile robot. Many of them have a three-position on/off switch which also selects the high/low volume. (See the video at **www.youtube.com/watch?v=0yvJ4ZiAZOY**.) Once again, some hacking is required to make room for the electronics. Specifically, I cut out the speaker's expanding bellows and replaced it with a fixed plastic ring about 1/2" wide, which holds a circular piece of perfboard (**Figure 4**). For 3.3V power, I connected to the internal lithium battery (just over 4V fully charged) through one silicon diode (~0.7V drop for a 1N4001). Like the RadioShack amp, I added a 1 mA LED and 330 ohm series resistor as a "speaking" indicator, and the same type of three-pin male servo cable connector.

BASIC STAMP 2 PROGRAM

```
' BASIC Stamp 2 program sings ROW ROW ROW YOUR
' BOAT on SP0-512
' {$STAMP BS2}
' {$PBASIC 2.5}
INPUT 0      'speaking line from 512
OUTPUT 7     '9600 true 8N1 data to 512
PAUSE 1000
SEROUT 7,84,[CR]    'initialize SP0-512
GOSUB waitspeak

main:  '  sing at speed 2
       'NOTE: timing spaces removed for magazine
       'listing; add multiple spaces between
       'words for timing delays
SEROUT 7,84,["[S2][G2]rowhh rowhh rowhh [A2]yer
[B2] boat jehnn [A2]dlee [B2]down [C3]thuh [D3]
stream",CR]
GOSUB waitspeak
PAUSE 450              ' pause for proper phrasing
SEROUT 7,84,["[G3]merrily [D3]merrily
[B2]merrily [G2]merrily [D3]lie ff[C3]is [B2]but
[A2]uh [G2]dream",CR]
GOSUB waitspeak
FOR B0=1 TO 5    'count from one to five
SEROUT 7,84,["[S3]    very ubble          B 0
equals             ",DEC B0,CR]
GOSUB waitspeak
NEXT
GOTO main

waitspeak:     ' wait subroutine
PAUSE 100                   ' wait 0.1 sec for
speaking line to go high
test:IF IN0=1 THEN test     ' wait until speaking
                            ' line goes low
PAUSE 100                 ' wait 0.1 sec after
                          ' speaking line goes low
RETURN
```

PICAXE 20M2 PROGRAM

```
 'PICAXE 20M2 program sings ROW ROW ROW YOUR
 'BOAT on SP0-512
#picaxe 20m2
#no_data
setfreq m16          ' 8 or 16 MHz required for
                     ' 9600 baud serial data
input c.5     'speaking line input from 512
output c.4    'output 9600 baud true 8N1 data
              ' to sp0-512
pause 1000  'wait to clear "ready" signal from
            ' 512
high c.4:pause 10'           manual 2 p.209 says
set pin high before sending true data
SEROUT c.4,T9600_16,(13) 'initialize 512 with a
                         'carriage return
GOSUB waitspeak

main:
SEROUT c.4,T9600_16,("[S2][G2]rowhh  [A2]yer
[B2] boat jehnn [A2]dlee [B2]down [C3]thuh [D3]
stream",13)
GOSUB waitspeak
PAUSE 450              ' pause for proper phrasing
SEROUT c.4,T9600_16,("[G3]merrily [D3]merrily
[B2]  [G2]merrily [D3]lie ff[C3]is [B2]but
[A2]uh [G2]dream",13)
GOSUB waitspeak
for b0=1 to 6 'lets count B0 from 1 to 6
SEROUT c.4,T9600_16,("[S3]the val you  uv  B
0 iz         ",#b0,"     you nitzz",13)
gosub waitspeak
next
goto main

waitspeak:    ' wait to speak subroutine
PAUSE 100
test:IF pinc.5=1 THEN waitspeak
PAUSE 100
RETURN
```

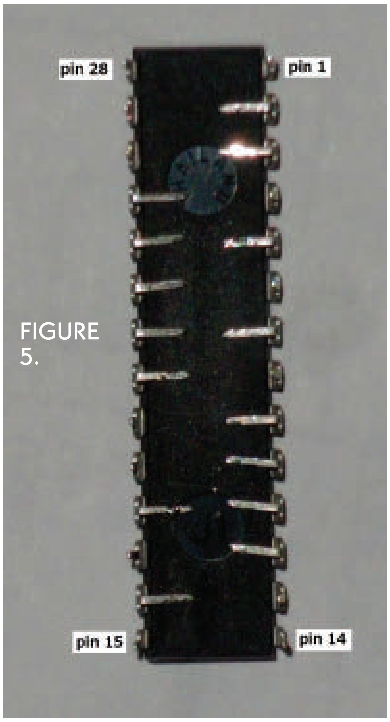


FIGURE 5.

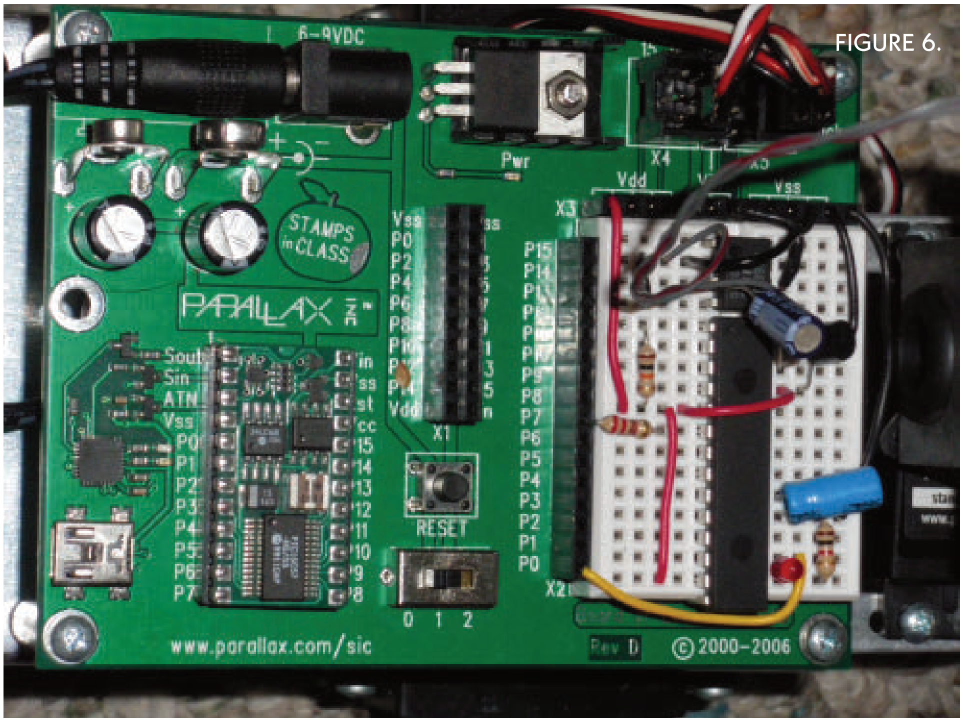


FIGURE 6.

Switched power connections and amplifier input on the PCB are easily traced with a multimeter. I removed the bottom three-conductor audio input cable entirely, and added two mounting screws while I was in there. RoboVoice audio output from pin 26 goes through a 10 µf cap into TWO connections on the amplifier PCB since it originally had stereo input. Just connect both together. You'll get reduced audio output if you only use one. BoeBot's breadboard area is just 17 rows long and nearly filled up by the 14-pin-long RoboVoice chip. That leaves just three rows of holes remaining after the chip is inserted. How do you add an LM386 audio amplifier then? CHEAT!

First, I gently folded the unused pins underneath the chip so they didn't even plug into the breadboard (**Figure 5**). Second, lower pins 14 and 15 are not used and hang over the edge of the breadboard in the breeze. (Not pretty, but it works!)

Those two cheats let you fit both chips into the breadboard and free up enough rows to make all the required connections (**Figure 6**).

A 3.3V regulator is required to power the 512. I found that the minimum-parts LM386 circuit — a 20X amplifier — drove a 16 ohm speaker with sufficiently loud volume. This was coupled to the RoboVoice directly through a 10 µF cap — no pot was used. Just for fun, I mounted the speaker on a servo to move it; the coffee cup horn definitely makes the sound louder and directional. An alternative to using the LM386 would be to use a hamburger amplifier speaker with only the RoboVoice chip plugged into the breadboard.

BASIC Stamp 2: Three simple connections use jumper wires or 10K resistors from the breadboard area to the adjacent pin headers. Pin 7 will use `SEROUT 7,84,["hello world",CR]` for speech; CR is a carriage return.

Arduino Uno: No shield is required since the female headers make it easy to connect. Digital pins 12 and 13 are adjacent to a ground connection; the male end of a servo extention cable plugs directly into these three connections. You will open a software serial port and use `Speak.println("hello world");` for speech. Note that `println` includes an automatic carriage return at the end.

PICAXE 20M2: To achieve 9600 baud, the 20M2 must be run at 8 or 16 MHz (default is 4 MHz); 16 MHz is recommended for robotics apps since the servo commands are calibrated only for 4 and 16 MHz. Nearly any I/O pins can be used, but note that C.6 is input only.

Per the manual, set the serial output pin high briefly before sending true data. Pin 4 uses `SEROUT 4,T9600_16,("hello world",13)` to invoke speech. That 13 is a carriage return.

A PICAXE makes an inexpensive terminal to experiment "live" with the 512 chip. A $4 PICAXE 20M2 can read a computer keyboard directly (no host computer) and sends the data to one of the 512 speaker/amp modules. There's a video at **www.youtube.com/watch?v=UhhDEhzF3PM** if you're interested.

With the addition of an LCD screen, you have the makings of a Speak & Spell type application. A video of the Arduino and BoeBot singing is at **www.youtube.com/watch?v=J5GMOAIQbJ4**.

Using RoboVoice

The RoboVoice can be used with

ARDUINO UNO SKETCH

```
// Arduino Uno sketch sings ROW ROW ROW YOUR
// BOAT on SP0-512

#include <SoftwareSerial.h>
#define txPin 13
#define spkPin 12
#define rxPin 11 // not implemented here
SoftwareSerial Speak(rxPin, txPin);

void setup() {
  // initialize serial communication with SP0512
  Speak.begin(9600);// set the data rate for the
                    // SoftwareSerial port
  delay(1000);        // pause 1 second at
                      // startup
  Speak.println(); // initialize 512 with one
                   // blank CR (println includes
CR)
  delay(500);
}

void loop() {
  // NOTE: timing spaces removed for magazine
  // listing; add multiple spaces between words
  // for timing delays
  Speak.println("[S2][G2]rowhh  rowhh  rowhh
[A2]yer  [B2] boat  jehnn  [A2]dlee [B2]down
[C3]thuh [D3] stream");
  waitspeak();  // call waitspeak function
  delay (500);  // pause for proper song spacing
  Speak.println("[G3]merrily [D3]merrily
[B2]merrily [G2]merrily  [D3]lie ff [C3]is
[B2]but [A2]uh [G2]dream");
  waitspeak();  // call waitspeak function
  delay (400);
}

void waitspeak()  // function: wait for 512 to
                  // finish speaking
{
  delay (100); // pause 0.1 sec for 512's busy
               // line to go high
  while (digitalRead(spkPin) == HIGH);
   //wait until 512's busy line goes low
  delay (100); // pause 0.1 sec after 512's
               // speaking line to goes low
}
```

many different processors. At power-up, it says "Ready" as the "speaking" LED lights up for a second. That's very helpful to know when testing your newly-built circuit for the first time; 9600 baud, 8N1, true serial data is sent to RoboVoice pin 6 where it accumulates in a buffer. Speech is triggered by either a carriage return (CR, typically a '13' character) or filling up the input buffer.

The latter should be avoided, since data may be lost while the chip is speaking. Your program should initially pause for one or two seconds while the RoboVoice speaks "ready," then send a CR (no data) to initialize communication.

When you first experiment with the chip, you will inevitably send some errant characters. As always, "garbage in, garbage out" wins, and this may lock up your chip. Remember, it is a microprocessor and bad data can crash it too. Reset the RoboVoice chip if it gives you the silent treatment when it should be chatty.

After each line of text is sent, your program needs a subroutine or function which waits 0.1 sec for the speaking line to go high; waits until it goes low (indicating that it has finished speaking); then waits 0.1 sec before sending more data.

The sample programs show this on the lines indicated. In my initial experiments, I sent two CRs after each line to trigger speech before learning those brief delays were necessary.

Send data (numbers) to the chip using the same formatting you would use to print or debug them. For instance, to make a BASIC Stamp 2 speak the value of variable B0, use this command:

```
SEROUT 7,84,["  variable B0
equals         ", DEC B0]
```

Master sending basic text ("hello world", 13) using serout or speak commands before trying the many control codes listed in the datasheet which change volume, speed, and pitch. Control codes are sent in capital letters inside square brackets with text. From the datasheet:

```
"[G2]This [A2]is [B2]a
[D3]test" will sing the words
"This is a test".
```

General TTS pronunciation using the built-in rules and algorithms is fairly good; in many cases, you'll do better by creative misspelling or using the phonemes listed in the datasheet mentioned previously.

There are control codes for various length pauses PA2-PA5, but I find it simpler to just add many spaces to make a longer pause — especially when empirically programming singing.

Some **abbreviated code** for all three processors is shown here to get you started. Have fun teaching your RoboVoice to speak and sing! **SV**

Arduino Remote Control Robot

by Gordon McComb

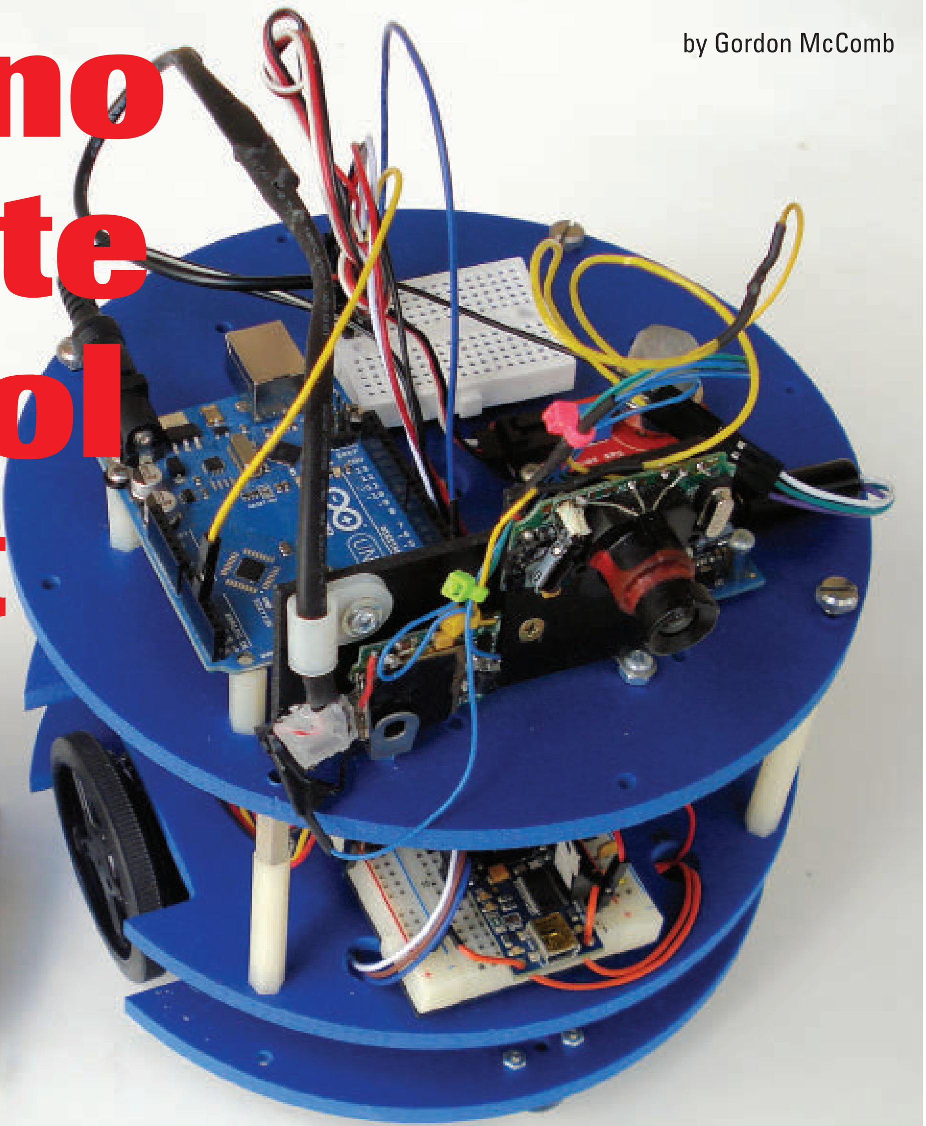


FIGURE 1. The Telebot is an ArdBot with a wire radio link and a wireless video camera capable of superimposing text over the picture. Beside the Telebot is the wireless remote control which uses its own Arduino.

A traditional robot is a mechanical contraption that operates under its own will. You upload a program to its brain, disconnect the cable, and let the beast roam free.

That's only one type of robot. Another is the remotely operated kind, where you guide the actions of the bot from afar. A link — wired or wireless — keeps builder and robot in constant communication. If the bot is in another room — or even another building, city, state, or country — a video camera can beam back pictures so that what the machine sees, its human operator can see, as well.

This is telerobotics: tele for distant, and robotics for, well, you know that part already. A telerobot is remotely controlled, though it may also contain autonomous functions that operate without human intervention. Some telerobots are commanded within the space of a living room, and some millions of miles away on alien planets.

In this article, you'll build the Telebot — a convergence of ordinary robot with remote control abilities. For the base, I've adapted the ArdBot chassis, described in the November 2010 through May 2011 issues of *SERVO*.

The ArdBot is a low cost and easily expandable base that uses an Arduino for a brain, and two continuous rotation servos for power.

Figure 1 shows a fully decked-out Telebot with handheld remote control and some additional hardware — a camera and video broadcaster — on top. I'll only talk about the remote control aspects here, as the add-ons are up to you.

Using Radio Waves to Control a Robot

There are numerous methods to control a robot. For this article, I'll demonstrate just one way using a radio link. For background purposes, here are a few other common schemes:

• **Wired link**. Connect a control panel consisting of buttons and switches directly to the robot's motors. The finished result is more like a motorized toy than a real robot, because you directly influence every action of the bot — it has no independent control.

• **Infrared remote control.** A universal TV remote can be used to send coded signals to a robot. A receiver mounted on the bot receives the signal, and a microcontroller commands the robot based on which buttons on the remote were pressed. Because the robot contains a microcontroller, it is capable of independent control in addition to human-influenced control. For example, you might press the button to have the robot go forward, but sensors on the bot will detect collision with something. The robot's own programming can reverse direction, without requiring any corrective action from you.

• **Line following.** In this type of control, a predefined line, dot, or other shape indicates where the robot is to go. Lines are usually drawn on paper, but it's also possible to interactivity draw the lines by using such techniques as a laser pointer, or even a large flat screen TV resting on its back.

• **Sound, light, or other sense.** These are all *indirect* control methods, where the robot is influenced by some sensory condition you manage. For example, you might build a robot that has a light detector that can be "steered" using a flashlight or handheld laser pointer. A *musicbot* might be able to differentiate tones played on a piano or guitar.

For the Telebot, I'm using the ZigBee standard of compact and affordable two-way radios.

Getting Started

ZigBee is a wireless data standard. Two ZigBee radios form a link for sending and receiving serial data over the air. The standard ZigBee is based on the IEEE 802.15.4 specification, intended for low speed, low power wireless data. ZigBee is only one of the technologies that uses the 802.15.4 specification — MiFi and WirelessHART are a couple others — but it's the one that's been most embraced by the robotics community.

The idea behind ZigBee is keeping costs low, so that wireless communications can be affordably built into consumer products such as light switches and home theater receivers. As a result, ZigBee data radios are among the least expensive. You can purchase a pair of them for under $50, depending on the feature set.

ZigBee operates at the 2.4 GHz (GigaHertz) band which is shared by a number of other wireless systems, including home Wi-Fi, cordless phones, and security video systems. Each ZigBee radio has a Data In pin (transmit) and a Data Out pin (receive). These two pins are referred to as DIN and DOUT, respectively. Though the typical ZigBee radio module has almost two dozen connection pins, other than a power and ground these are the only wires needed to establish an over-the-air serial link. A self-contained module that conforms to the ZigBee standard — complete with integrated antenna and mounted on an carrier board — is shown in **Figure 2**.

There are numerous variations among ZigBee radios. If you're interested in learning more, check out the Wikipedia page on ZigBee for more information. Radios of different sets aren't compatible with one another, so be sure to get two modules that match. Here's a basic rundown of some of the variations:

• Basic ZigBee is often referred to simply as 802.15.4 (the IEEE standard that defines it), Series 1, or S1.

• The "extended" ZigBee is better known as ZB. It uses a different and more complicated communications protocol, among other differences.

• As a side benefit of its low cost nature, many ZigBee modules designed for hobby and experimental use also include a number of handy data acquisition features, highly useful for robotics. For example, the module might have one or more of its own analog-to-digital (ADC) inputs, as well as multiple digital input/output pins. I won't be talking about these features here, but you should know they're available if you need them.

FIGURE 2. An XBee IEEE 802.15.4 (Series 1) radio on a Parallax carrier board. The board provides regulated 3.3V to the XBee.

The Telebot robot uses a pair of low cost XBee Series 1 modules. XBee is a popular brand of ZigBee RF radios that support the 802.15.4 protocol. The form factor — the physical layout — of the XBee radios is widely supported with various adapters that allow you to plug them directly into solderless breadboards. Since the transmission range does not need to be extensive, the low power one milliwatt (1 mW) modules are more than adequate.

XBee modules use connector pins that are set 2 mm apart. In order to plug the XBee module into a solderless breadboard, you must first connect it to a carrier or adapter. These carriers come in various styles with a variety of features. Series 1 modules are simple to use. If you don't mind the default 9600 baud communications speed, they provide out-of-the-box factory settings for quick and easy setup. While you can always modify the factory settings for such things as communications speed, channel, and ID number (so more than two XBees can link at the same time), none of this is required if you're only needing to establish a basic wireless serial link between two nodes (points).

Setting Up the Telebot Remote

Telebot uses two Arduino controllers. A main Arduino on the Telebot provides the primary functions of the robot such as motor control, and a remote control sends wireless commands to the Telebot. The remote uses any of several sensors to provide actuation; I'll be demonstrating the use of a five-position switch to control forward/back and right/left motion. A separate Arduino is used to process these sensors. Both the Telebot and the remote control are outfitted with XBee radios for transmitting the commands from the remote.

The remote for the Telebot consists of an Arduino Uno development board, prototyping shield with mini solderless breadboard, battery, Series 1 XBee radio, and a Parallax five-position switch. The five-position switch is literally five switches in one, arranged to provide up/down/left/right navigation. (The fifth switch is a center pushbutton.) Each of the five switches has its own pull-up resistor, and is interfaced to the Arduino with only a simple wire connection.

Figure 3 shows a drilling and cutting layout for the handheld remote. The Arduino is powered by a nine volt battery which you can hold in place with a battery clip or piece of double-sided foam tape. **Figure 4** shows the five-position switch.

Alternatives to using a multi-axis switch include an accelerometer and tilt compensated compass.

Begin construction of the Telebot remote by cutting and drilling the base plate in **Figure 3**. Mount the Arduino to the plate using a set of four 1/2" long nylon standoffs and 4-40 machine screws. Use 4-40 x 1/2" flat head screws on the underside of the plate, and 4-40 x 3/8" screws to mount the Arduino board to the standoffs. If using metal screws, add a plastic washer to prevent any possible shorts.

Attach a standard protoshield over the Arduino. If the shield does not already have a mini breadboard on it, attach one using double-sided foam tape (most breadboards have the tape already applied).

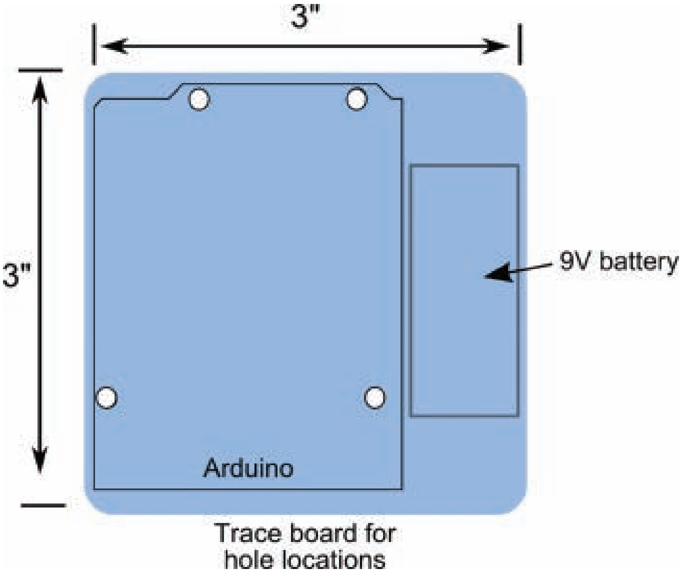


FIGURE 3. Layout guide for the remote baseplate. Use the Arduino and (optional) Devantech CMPS10 compass for marking holes for drilling.

Listing 1 — Telebot_Transmit.ino.

```
#include <SoftwareSerial.h>

// Pin connections for switch (includes power and ground)
const int pwrPin = 7;
const int gndPin = 4;
const int upBttn = 5;
const int crBttn = 6;
const int rtBttn = 8;
const int dnBttn = 9;
const int ltBttn = 10;

// Pin connections for XBee radio
const int xb_rx = 2;
const int xb_tx = 3;
SoftwareSerial Xbee(xb_rx, xb_tx);

int keyDelay = 300;          // Limit keypress to one per 300ms
long previousMillis = 0;    // Stores last time switch updated

void setup() {
  Xbee.begin(9600);

  // Set power to 5-position switch
  pinMode(pwrPin, OUTPUT);
  digitalWrite(pwrPin, HIGH);
  pinMode(gndPin, OUTPUT);
  digitalWrite(gndPin, LOW);
}

void loop() {
  unsigned long currentMillis = millis();

  // Prevents bounce
  if(currentMillis - previousMillis > keyDelay) {
    previousMillis = currentMillis;
    if(digitalRead(ltBttn) == LOW)
      Xbee.print("l");

    if(digitalRead(rtBttn)== LOW)
      Xbee.print("r");

    if(digitalRead(upBttn) == LOW)
      Xbee.print("u");

    if(digitalRead(dnBttn) == LOW)
      Xbee.print("d");

    if(digitalRead(crBttn) == LOW)
      Xbee.print("c");

  }
}
```

All three of the control sensors are linked by way of the Arduino to a Series 1 XBee radio. The wiring diagram is shown in **Figure 5**. The radio is mounted on a 22-pin carrier available from Parallax. Remember that XBees are designed for 3.3 volt operation. I'm using a carrier made for either 3.3V or 5V operation. The carrier has its own built-in 3.3V regulator, so you can use the Arduino's 5V supply.

Why not just use the Arduino's 3.3V power supply? It's a matter of power consumption. The 3.3V regulator on the Arduino Uno and similar boards is limited to providing 50 mA of current. That's about what the XBee uses when transmitting — that's too close for comfort.

Though the XBee works on 3.3 volts, the input and output pins on the module are 5V tolerant. That means they can be directly connected to a 5V device — like the Arduino Uno — without the need for current-limiting resistors or level-shifting electronics.

The wiring diagram for the five-position switch is shown in **Figure 6**. The switch comes on an eight-pin breakout board that's a bit oversized, but will still fit onto the breadboard with the XBee module. Programming code is provided in **Listing 1**. The sketch sends out a single byte character, indicating the direction of the switch press: *u* for up; *d* for down; and so forth. The completed remote control is shown in **Figure 7**.

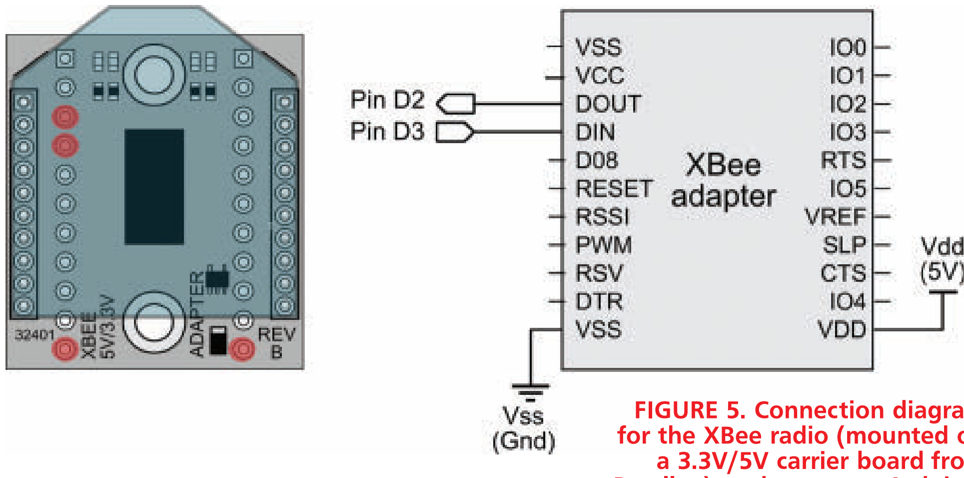
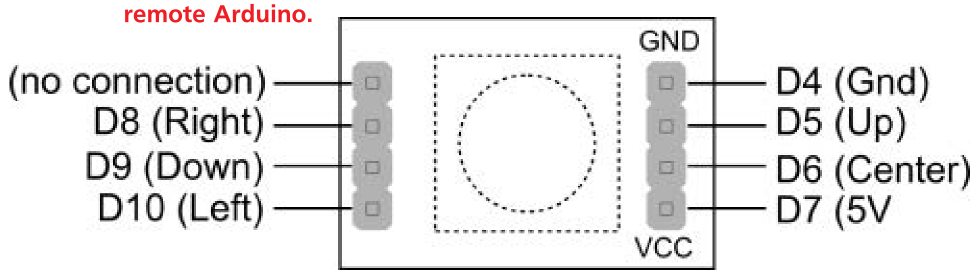


FIGURE 5. Connection diagram for the XBee radio (mounted on a 3.3V/5V carrier board from Parallax) to the remote Arduino.

FIGURE 6. Wiring diagram for the five-position switch to the remote Arduino.



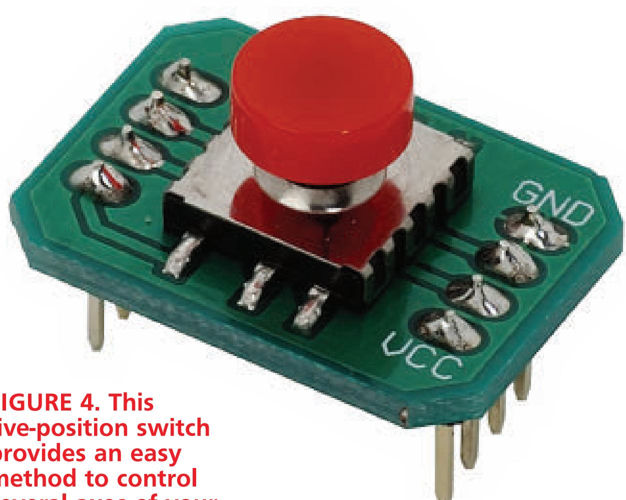


FIGURE 4. This five-position switch provides an easy method to control several axes of your remote controlled robot.

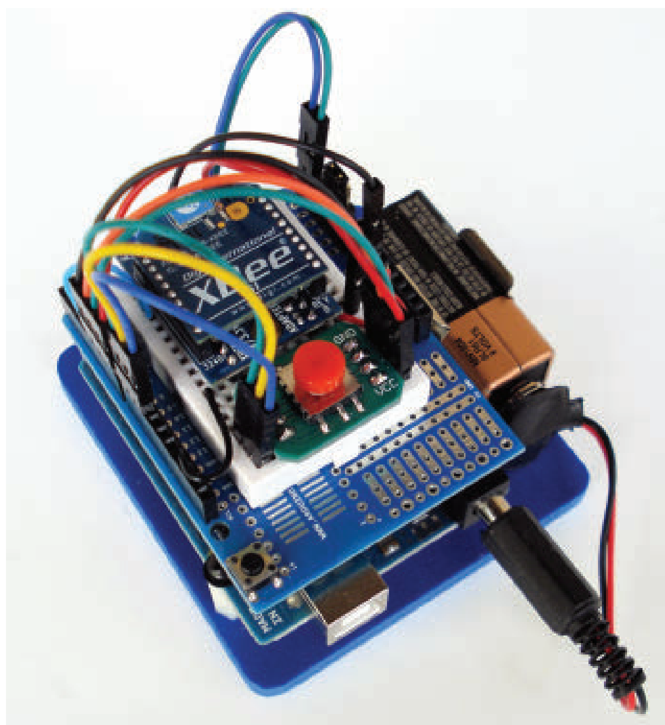


FIGURE 7. The completed remote control with Arduino, protoshield with mini breadboard, nine volt battery, XBee radio, and five-position switch. I have a couple of extra wires on my prototype, since it has the tilt-compensated compass (not described here because of space reasons).

Adding an XBee Receiver to the Telebot

To turn the ArdBot into the Telebot, you only need to add an XBee module to receive the signals from the remote control unit. To save space on the breadboard atop the robot, I've elected to use a SIP-style XBee carrier — also available from Parallax and shown in **Figure 8**. This particular carrier board lacks mounting holes, but it can be attached using a cable tie threaded through two holes.

Electrical connection requires merely four wires (see **Figure 9**): two for power and one each for data receive and transmit.

As noted, when using two Series 1 XBee radios, you can keep the default factory settings and the two radios in your link will automatically know how to communicate with one another. Transfer speed is 9600 baud, which is plenty fast enough for the kind of simple data used for the Telebot. A speed of 9600 baud is over 900 bytes per second, and the most data sent from the remote is five bytes at a time.

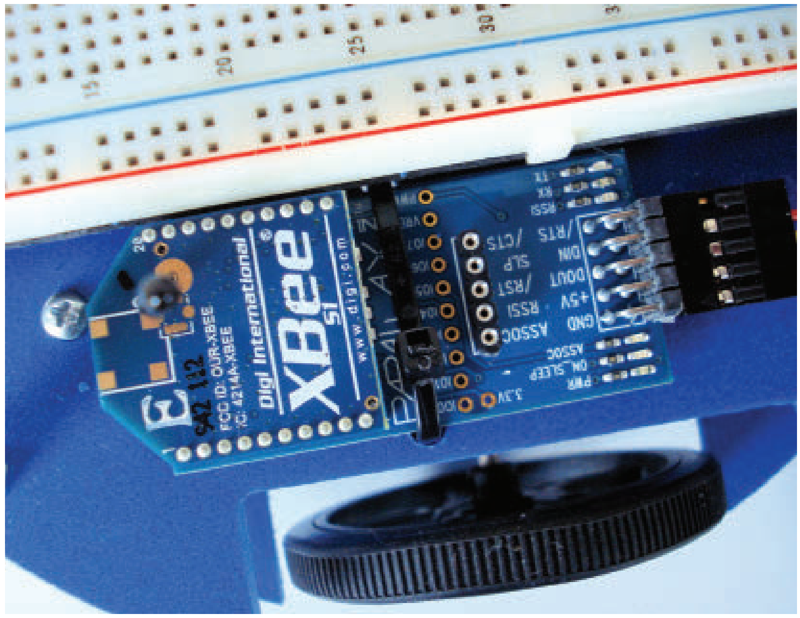


FIGURE 8. The XBee radio on the Telebot is mounted on a SIP adapter, and is secured to the base of the robot using an ordinary cable wrap.

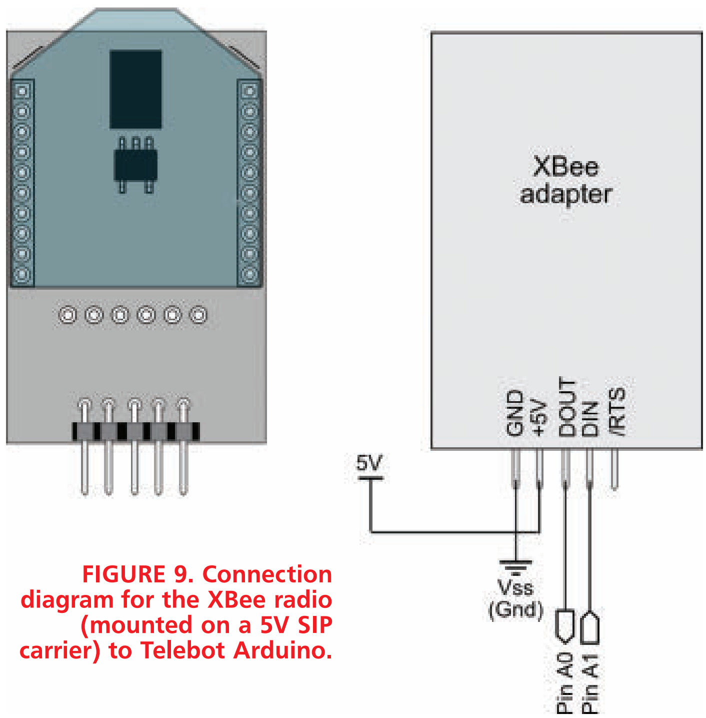


FIGURE 9. Connection diagram for the XBee radio (mounted on a 5V SIP carrier) to Telebot Arduino.

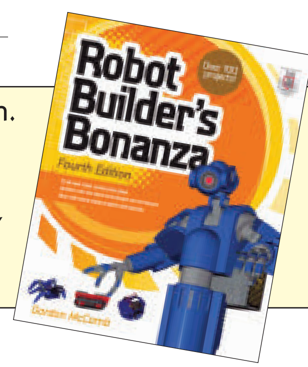
Sources

Budget Robotics
ArdBot chassis kit
www.budgetrobotics.com

Parallax
XBee and XBee adapter boards, continuous rotation servos, five-position switch
www.parallax.com

Pololu
Arduino Uno, continuous rotation servos
www.pololu.com

SparkFun
Arduino Uno, continuous rotation servos, XBee modules
www.sparkfun.com



Listing 2 shows how to interface the XBee receiver to the serial motor controller used on the Telebot. Note the use of the SoftwareSerial and Servo object libraries — both of which come with the Arduino IDE software. Points of interest in the sketch include:

- The *setup()* routine starts three serial objects (one for the XBee, two for the motor controller).
- The *readXBee* function continuously reads the XBee receiver. If there's data, it processes it using the *controlMotor* routine.
- In the *controlMotor* routine, the bytes received from the XBee are compared against the five expected single-letter values. Each value has a corresponding motor routine. For example:

```
 case ('u'):
forward();
```

This runs both motors forward when receiving the *u* (up) signal.

Last Data to Receive

The Telebot provides numerous avenues for customization and enhancement. Add a video camera as I've done for the prototype, and you can beam back pictures as you drive your robot through your house or office. In conjunction with the camera, I added a Parallax Backpack module to act as a video text overlay. A third Arduino atop the Telebot collects data from a series of sensors mounted nearby — temperature, humidity, light, and so on. The data is captured and added as text over the video that's transmitted back to me.

Even if you don't use picture feedback, your Telebot will give you hours of experimentation. Remember that just because you control the robot with a joystick, that doesn't mean the Telebot has to be completely reliant on you. Add some sensors to allow it to navigate on its own. Steer it into a corner, then see how well it can find its way out! **SV**

www.servomagazine.com/index.php?/magazine/article/november2012_McComb

Discuss this article in the SERVO Magazine forums at http://forum.servomagazine.com

Listing 2 — Telebot_Receive.

```
#include <SoftwareSerial.h>
#include <Servo.h>

Servo servoLeft;                // Define left servo
Servo servoRight;               // Define right servo

#define xbeeRx     A0            // DOUT to A0
#define xbeeTx     A1            // DIN to A1

SoftwareSerial Xbee (xbeeRx, xbeeTx);    //After motor serial

void setup() {
  servoLeft.attach(10);         // Left servo to pin 10
  servoRight.attach(9);         // Right servo to pin 9
  Serial.begin(9600);
  Xbee.begin(9600);
  delay(200);
  pinMode(13, OUTPUT);         // Show ready status
  digitalWrite(13, HIGH);
}

void loop() {
  readXbee();
  delay(50);
}

void readXbee() {
   if(Xbee.available()) {
     char val = Xbee.read();
     Serial.println(val);
     controlMotor(val);
  }
}

void controlMotor(char val) {
  switch (val) {
    case ('c'):
      stopRobot();
      break;
    case ('u'):
      forward();
      break;
    case ('d'):
      reverse();
      break;
    case ('r'):
      turnRight();
      break;
    case ('l'):
      turnLeft();
      break;
  }
}

// Motion routines for forward, reverse, turns, and stop
void forward() {
  servoLeft.write(0);
  servoRight.write(180);
}
void reverse() {
  servoLeft.write(180);
  servoRight.write(0);
}
void turnRight() {
  servoLeft.write(180);
  servoRight.write(180);
}
void turnLeft() {
  servoLeft.write(0);
  servoRight.write(0);
}
void stopRobot() {
  servoLeft.write(90);
  servoRight.write(90);
}
```

Review: The ArduPilot 2.5
Autopilot Board and Mission Planner Software

by Bryan Bergeron

Mission Planner software showing earth Google location of the autopilot, artificial horizon, and a cluster of information gauges.

www.servomagazine.com/index.php?/magazine/article/november2012_Bergeron

Discuss this article in the *SERVO Magazine* forums at http://forum.servomagazine.com

If you're an Arduino fan and want to build semi-autonomous planes, copters, or land vehicles, then this review of the ArduPilot 2.5 from DIY Drones is a must-read. As you'll see, the graphical Mission Planner software makes this powerful autopilot both plug-and-play and yet fully accessible for customization. Be forewarned, however, this system is so much fun and so deep that's it's highly addictive.

Introduction

Thanks in part to affordable frames and components, and also in part to enthusiastic support from selfless open source developers, building a DIY quadcopter is no longer an arduous, expensive proposition. Quadcopter frames, motors, and electronic speed controllers (ESCs) are commodity items that can be cherry-picked to your speed and payload constraints, and of course, your budget. For robotics enthusiasts, the only remaining challenge in the world of quadcopters is developing the perfect microcontroller-based autopilot.

As with microcontroller platforms in general, there's a world of possible autopilot options available — from the $15 AVR-based minimalist controllers offered by HobbyKing and the $120 Parallax Propeller-based HoverFly Open Board (which I discussed in the September 2012 issue), to the $200 ArduPilot 2.5, featured here. These three autopilots target very different audiences. And, of course, everyone has their favorite microcontroller platform.

A major distinction of the ArduPilot 2.5 (referred to hereafter as simply 'ArduPilot') and the accompanying freely available Mission Planner software is that it supports true hands-off, autonomous flying. The system allows you to control a quadcopter, airplane, rover, or other vehicle without ever using a traditional R/C transmitter-receiver system.

As long as you can use a mouse and keyboard, you can direct the autopilot and it will take care of the details. The platform also leverages the largest, most active user and developer base for civilian, non-commercial quadcopter research and development.

The Hardware

Figures 1 and **2** show the latest version of the ArduPilot. As you can see from the photos, the board is cleanly laid out with the input and output pads clearly identified, with enough space around the corner holes to actually mount the board. The board is packed with sensors and LED status lights. The 41 mm x 66 mm ArduPilot is packed with the latest generation three-axis gyro, accelerometer, magnetometer, and barometer. The barometer isn't for forecasting the weather, but for measuring altitude.

There's 4 MB of Flash memory on board, and Atmel's ATMEGA2560 for the heavy lifting. There's also AMEGA32U-2 chips for USB functions. What's more, the board is designed to handle a variety of extra sensors and peripherals, such as GPS receivers and ultrasonic range finders. An ultrasonic range finder is best for low level, relatively high accuracy altitude measurements. The barometer is best for estimating elevation above about 10 meters. Adding standard sensors is as easy as attaching the leads to the ArduPilot and then clicking a few buttons within the Mission Planner software.

For this review, the only external peripheral I tested

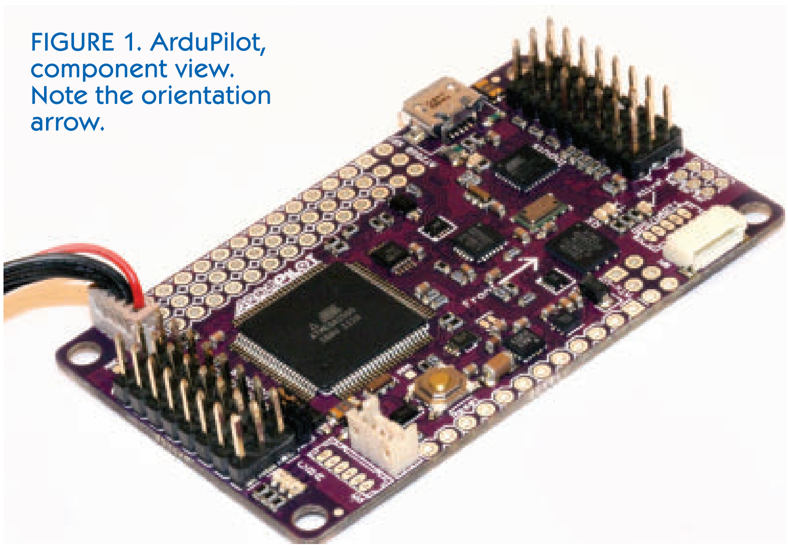


FIGURE 1. ArduPilot, component view. Note the orientation arrow.

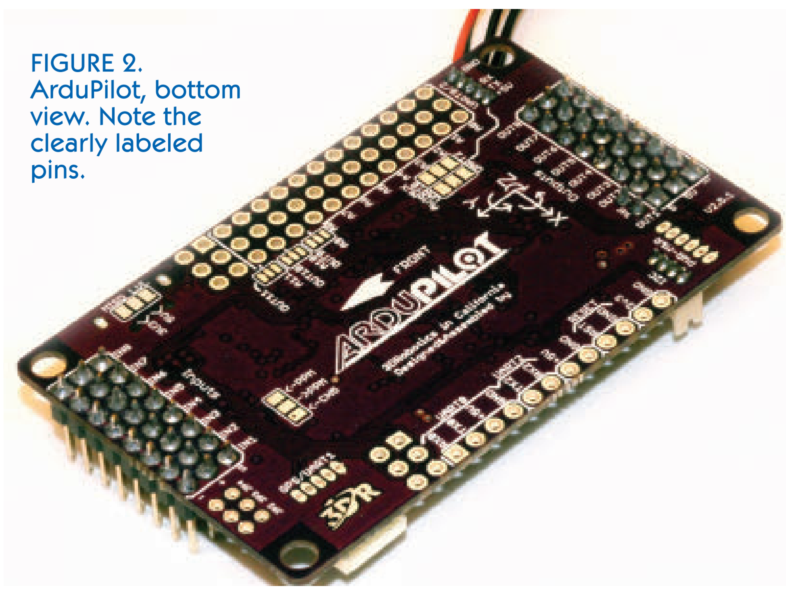


FIGURE 2. ArduPilot, bottom view. Note the clearly labeled pins.

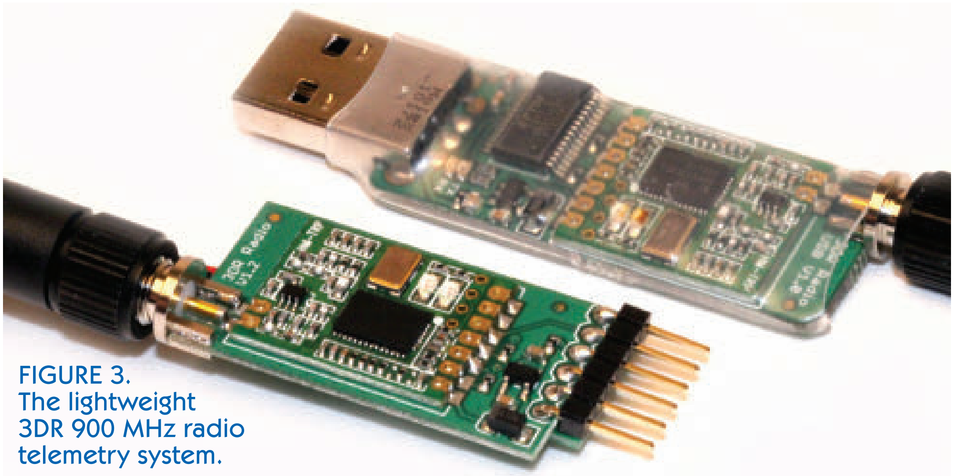


FIGURE 3. The lightweight 3DR 900 MHz radio telemetry system.

FIGURE 4. Close-up of the power distribution pads for the Q450 fiberglass frame.

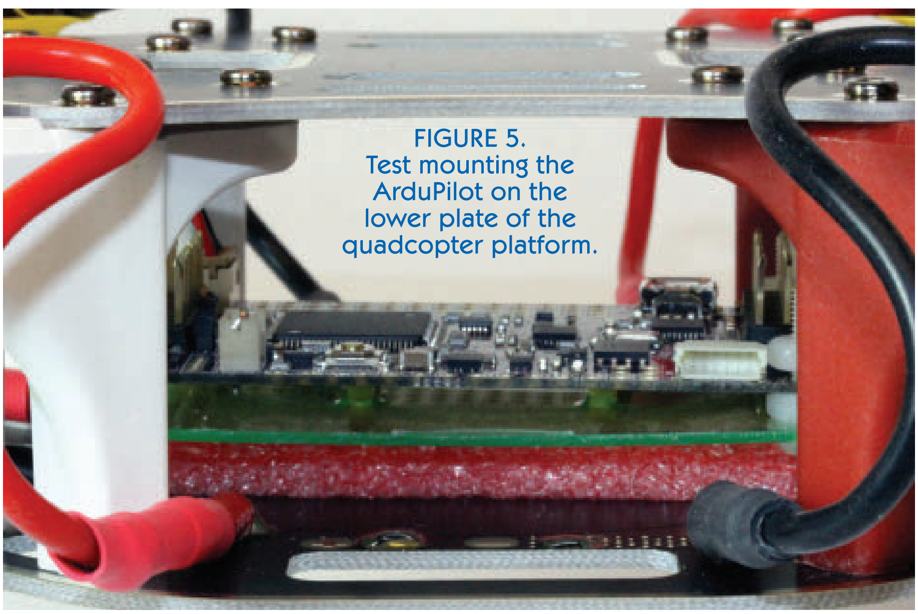


FIGURE 5. Test mounting the ArduPilot on the lower plate of the quadcopter platform.

was the 3DR 900 MHz radio telemetry system from DIY Drones (see **Figure 3**). The $75 system is sold as an XBee killer, with a smaller footprint, lighter weight, and duplex communications range of about one mile. One unit plugs into the USB socket of your laptop and the other unit attaches — via cable — directly to the board. It provides a virtual USB connection between the board and your PC with data rates of about 250 kbps. The top unit in **Figure 3** plugs into the USB port on a laptop, while the lower unit attaches — via cable — to the ArduPilot.

The Quadcopter Platform

The ArduPilot and Mission Planner software are an amazing duo, but they need a platform to show off their capabilities. DIY Drones sells several quadcopter frames, designed to carry from three to six engines. However, in the spirit of open source and DIY, I opted for a Q450 fiberglass frame with integrated PCB from HobbyKing. The white and red frame was $18 and about the same for shipping from Hong Kong — still, a bargain at $36.

The Q450 was easy to set up and wire, thanks to the integrated power distribution pads on the PCB shown in **Figure 4**. I attached a LiPo battery using 4.5 mm bullet connectors to the pads on the lower left of the **photo**. Similarly, I soldered the four pairs of power leads to the ESCs using 3.5 mm bullet connectors, then to the four sets of pads at the base of each arm. Total time: about 10 minutes.

In addition to the cost of the frame, there were four 30A ESCs ($13 each), four 1,000 KV motors ($14 each), a set of propellers (two clockwise and two counterclockwise; $2 each), and four collets ($1 each) to hold the propellers on the motor shafts. I purchased everything from HobbyKing. I also supplied several 4,400 mAh LiPo batteries, a smart

LiPo battery charger, a Spektrum DX6i, six-channel R/C unit, and a handful of tie wraps from my existing inventory.

It's a significant investment if you're starting from ground zero, but don't give in to the urge to skimp on the R/C unit. Saving a few dollars up front won't make you feel any better when you see your quadcopter dash over the horizon out of control because the R/C system failed.

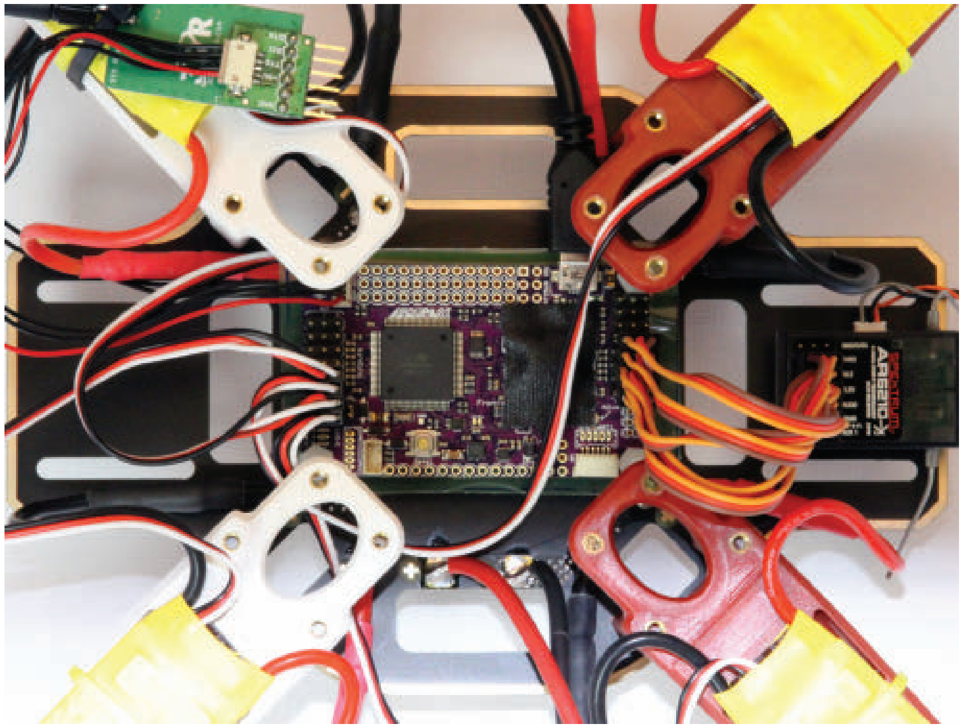


FIGURE 6. Wiring the output to the ESCs (left) and the input from the radio receiver (right).

FIGURE 7. Use a pin extractor on three short servo extension cables to limit the possibility of a ground loop from multiple 5 VDC supplies in parallel.

The entire frame build time was about an hour, with half of that time devoted to mounting the ArduPilot to the lower platform of the Q450 (see **Figure 5**). Frame vibration is always a concern when using an autopilot because it adds noise to the sensors. My workaround is to first mount the board onto a soft plastic platform using nylon bolts and silicon vibration-damping washers. Then, I put that combination on a 1/4 inch layer of foam which I tie wrap to the frame. The combination provides a soft cushioned ride at minimum weight expense.

After connecting the motors to the ESCs, I wired in the ArduPilot, as shown in **Figure 6**. Note the black square of gaffer's tape on the right side of the board in the **figure**. There is a small square of cloth between it and the surface of the barometric pressure sensor. This 'bandaid' of sorts allows the barometric pressure sensor to operate free from the effects of the air rushing by. This is a non-issue on quadcopters in which the autopilot board is shielded by a windproof canopy.

Figure 6 shows the ESCs connected directly to the ArduPilot. That is, the board is being supplied by four 5 VDC supplies in parallel. To minimize the possibility of a ground loop, you can cut the power wire (red) on three of the four ESCs, or non-destructively remove the center power pin from three of the four ESC connectors.

An alternative approach is to use a pin extractor to remove the power wire from three short servo extension cables (see **Figure 7**), leaving one ESC to provide the ArduPilot with 5 VDC. There's a bit of a weight penalty associated with using the extension cables but when you rip an ESC to shreds in a crash, it's much easier to replace the ESC with one on hand if you don't have to worry

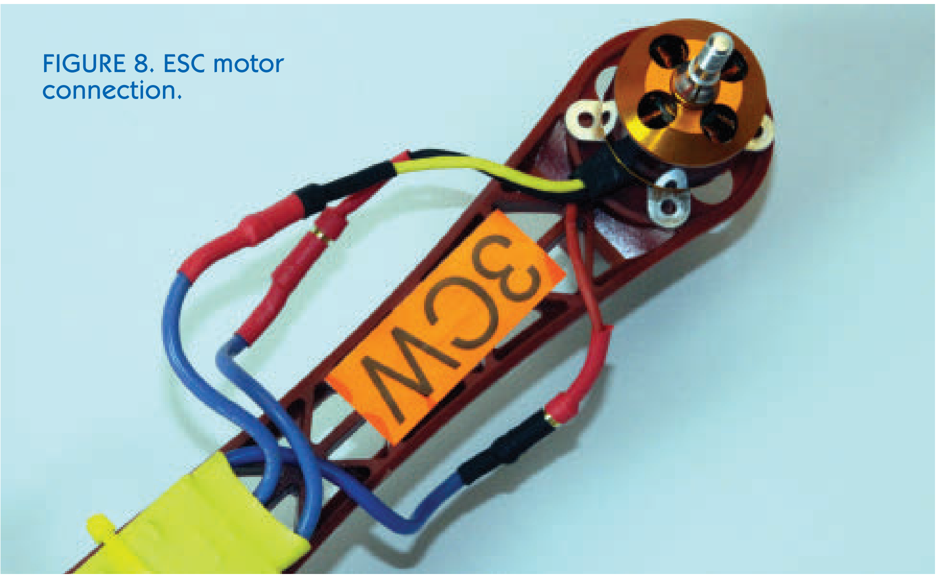


FIGURE 8. ESC motor connection.

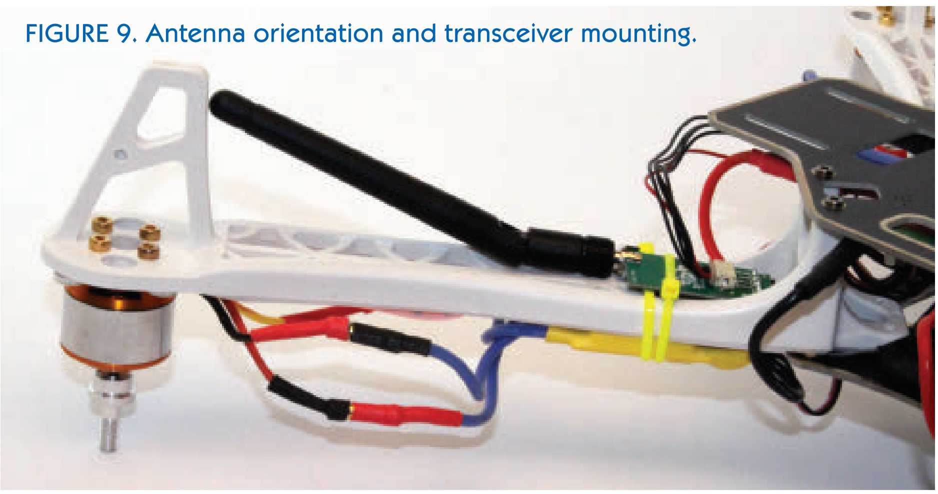


FIGURE 9. Antenna orientation and transceiver mounting.

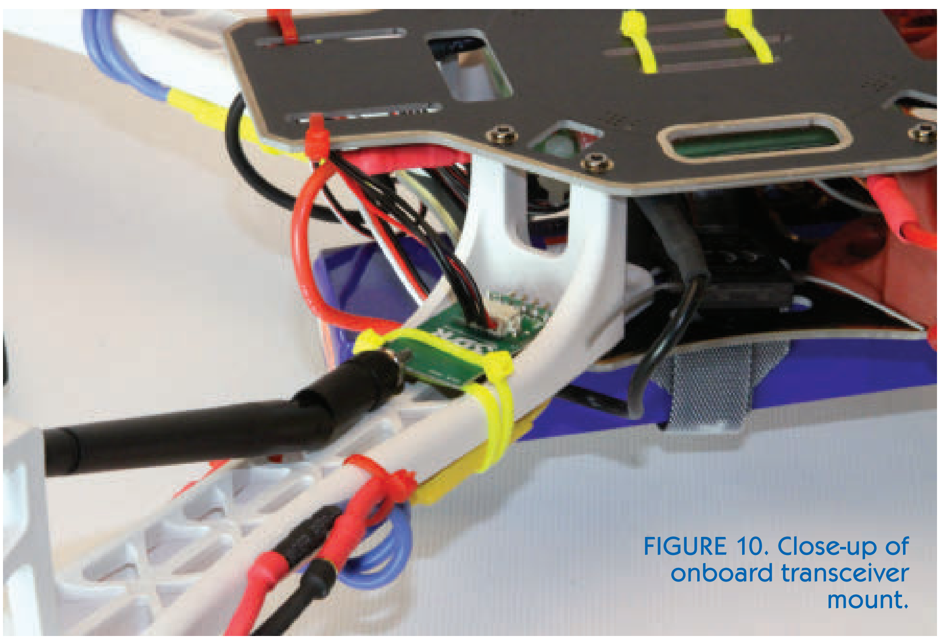


FIGURE 10. Close-up of onboard transceiver mount.

about whether it supplies power to the autopilot.

Next, I connected the ESCs to the motors using 3.5 mm bullet connectors as in **Figure 8**. Note that the wires aren't yet secured to the frame because I haven't verified the proper direction of rotation for each motor. That comes later, when power is applied to the system. I also labeled each arm of the drone by channel number on the ArduPilot and by direction of prop rotation.

For example, the ESC driving the motor in **Figure 8** is connected to the third channel on the autopilot board, and the prop spins clockwise (CW). The labeling helps when you're reassembling a quadcopter after a crash in the field. If you happen to reverse a prop, then odds are the quadcopter won't make it more than a foot or two above the ground before it takes a dive.

Because the Q450 frame has limited ground clearance, mounting the onboard 900 MHz antenna vertically for omni-directional coverage wasn't an option. Instead, I mounted the unit under one of the arms, nestled as close to the main body as possible (refer to **Figures 9** and **10**). If you decide to follow this example, then nudge the ESC on the opposite arm a few cm away from the body. This will help bring the quadcopter back into balance.

Setup and Testing

With the mechanical build out of the way, the next step is to connect the ArduPilot to a PC running the freely available Mission Planner software, using the mini USB port. The 5 VDC from the PC's USB port won't power the ESCs or motors — for that, you'll need a battery or power supply. I prefer the latter, because I can quickly disconnect the power and monitor the current drain. See **Figure 11** for my setup.

Notice that the propellers are not attached at this time. You're going to be starting and stopping the board,

and working with the input and output connectors on the ArduPilot. This translates to ample opportunity to slice a finger or two. Keep the props off until the very end, and don't forget to balance the props — there are lots of examples on YouTube if you're new at this.

Firmware setup of the ArduPilot is a breeze, thanks to the menu-driven Mission Planner. **Figure 12** shows the first step in the menu-driven setup, calibrating the R/C unit mapping onto the ArduPilot. I could have dropped into Arduino source code and set these parameters manually, but there was no need. As you can see in **Figure 12**, the next steps are listed in order, starting with flight modes and ending with the Planner. Setup is painless and straightforward, requiring all of five minutes to complete.

Time to Get Physical

After setup and testing, including checking and — if necessary — correcting prop rotation direction, it's time to drop in a battery, boot the laptop, and take it for a spin. **Figure 13** shows the quadcopter ready to go. Note that there's no on-off switch on this creature. Once you plug in the battery, the ESCs issue various beeps and buzzes, so you'd better stand back. There is an arming sequence which requires you to hold the throttle stick in a certain position for four seconds. A constant red light signifies the ArduPilot is armed and ready to go.

Figure 14 shows one of many views available within the Mission Planner. This view shows the artificial horizon in the upper left, geo-location on the right, and basic sensor information from the ArduPilot on the lower left. In this example, ground speed is 0 (the quadcopter was on my workbench), while altitude is shown as 2.14 feet.

In addition to a realistic cockpit experience, you can look at the raw sensor data in real time, as shown in **Figure 15**. In addition to instantaneous graphical displays of roll, pitch, and yaw dials, you can see the plots of data from the accelerometer and gyro over time.

I probably shouldn't reveal this, but at the time I

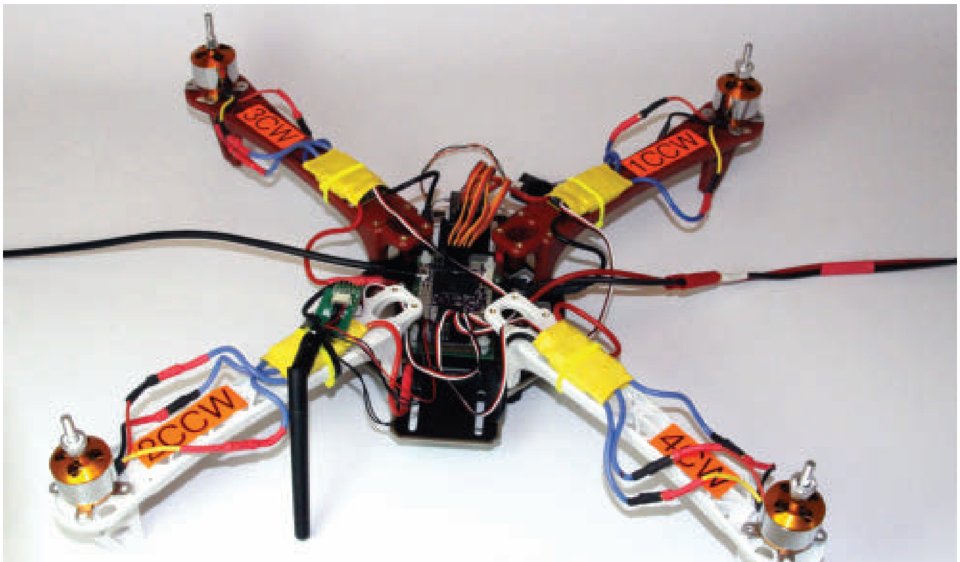


FIGURE 11. ArduPilot setup with external power and USB connectivity to the Mission Planner software.

FIGURE 12. R/C controller mapping within the Mission Planner system.

FIGURE 13. ArduPilot mounted within the quadcopter frame, ready to go.

FIGURE 14. One of several views from within the Mission Planner software.

finished setting up the ArduPilot and quadcopter frame, it was pouring down rain outside. So, I tried the quadcopter in my bedroom. At first, I held the copter overhead while an assistant armed and then worked the controls. After I was certain the craft responded as it should have, I dipped and rotated the craft, paying attention to the motors working to stabilize it.

When I was satisfied with that, I disarmed the copter and placed it on the bed. I had the craft make a few short bounces and then a few circles. There was no problem with the flight controls or the real time data streaming into my laptop.The next day with the sun shining and with several fully charged LiPos in tow, I brought the system to a park for a few test runs. The board performed superbly. I had to reduce the sensitivity of the system so that it performed well in modest wind.

The procedure took all of five minutes — on site — using the menu system within the Mission Planner software. I didn't test the full one mile range of the telemetry system, but it worked flawlessly at about a quarter mile radius — the maximum distance I was comfortable with flying the craft manually.

Waypoints

Figure 16 shows the Waypoints interface to the Mission Planner software. Setting waypoints is as simple as zooming into a place on Google Maps, then defining a flight path with a few clicks of your mouse. In this example, I zoomed in on Boston's Fenway Park, with a flyover that surveyed most of the field. In theory, I could launch my quadcopter outside the ballpark, have it hop over the bleachers, make the rounds, perhaps loiter over home base for effect, and then exit over the top of the bleachers. Of course, after doing so, I'd be writing reviews by hand on tissue paper in a cell.

Although the forums are full of accounts of waypoint success stories, I didn't check out this function first-hand. The new external GPS receiver designed to plug into the ArduPilot wasn't available at the time of my review. Even so, the Mission Planner software interface was fascinating and solid. Obviously, I have a GPS on order.

FIGURE 15. Raw sensor output data.

Closing Thoughts

In short, the new ArduPilot is a winner. The feature-packed hardware leverages a fantastic piece of freely available software, and has a strong user base and active forum behind it.

The diminutive and lightweight powerhouse of sensors and I/O ports is — to my knowledge — a combination unparalleled in the world of affordable autopilots for quadcopters. If you enjoy coding the Arduino and want the option of using a fantastic graphical user interface to create the code for you, then this is your card.

As I noted earlier, the problem with this system is that it's deceptively simple to work with. As you start to peel away at the onion, you'll soon discover hundreds of options that you can explore on your computer and then in the field — and that's what robotics is all about. There's no substitute for hands-on experimentation, and the ArduPilot manages to make the task approachable for the novice, and yet challenging for the seasoned experimentalist. **SV**

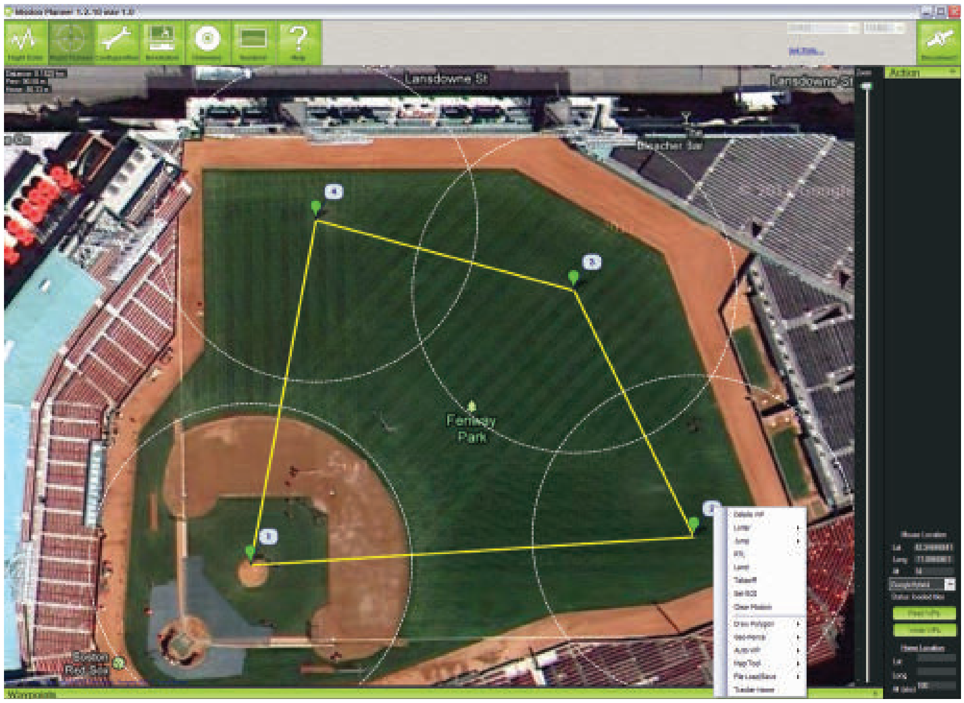


FIGURE 16. Waypoint authoring tool showing a four-leg jaunt in Boston's Fenway Park.

The *SERVO* Webstore

Attention Subscribers ask about your discount on prices marked with an *

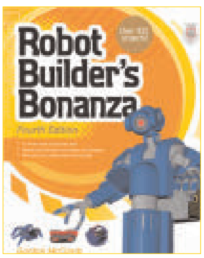
CD-ROM SPECIALS

NEW RELEASE

ROBOTICS

Robot Builder's Bonanza, Fourth Edition

by Gordon McComb

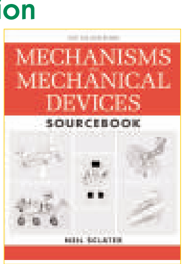


Robot Builder's Bonanza, Fourth Edition includes step-by-step plans for a number of motorized platforms. The book is described as a compendium of robotics topics, containing more than 100 projects, including 10 robot designs new to the fourth edition. These modular robots are low cost, and are made to be reproduced by readers with no training in mechanical construction.

$29.95*

Mechanisms and Mechanical Devices Sourcebook 5th Edition

by Neil Sclater



Fully revised throughout, this abundantly illustrated reference describes proven mechanisms and mechanical devices. Each illustration represents a design concept that can easily be recycled for use in new or modified mechanical, electromechanical, or mechatronic products. Tutorials on the basics of mechanisms and motion control systems introduce you to those subjects or act as a refresher.

Reg $89.95 Sale Price $79.95

Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists

by Dustyn Roberts

In *Making Things Move: DIY Mechanisms for Inventors, Hobbyists, and Artists*, you'll learn how to successfully build moving mechanisms through non-technical explanations, examples, and do-it-yourself projects — from kinetic art installations to creative toys to energy-harvesting devices. Photographs, illustrations, screenshots, and images of 3D models are included for each project.

$29.95*

Build Your Own Humanoid Robots

by Karl Williams

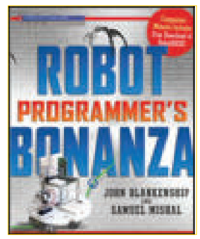
GREAT 'DROIDS, INDEED!

This unique guide to sophisticated robotics projects brings humanoid robot construction home to the hobbyist. Written by a well-known figure in the robotics community, *Build Your Own Humanoid Robots* provides step-by-step directions for six exciting projects, each costing less than $300. Together, they form the essential ingredients for making your own humanoid robot.

$24.95*

Robot Programmer's Bonanza

by John Blankenship, Samuel Mishal



The first hands-on programming guide for today's robot hobbyist!

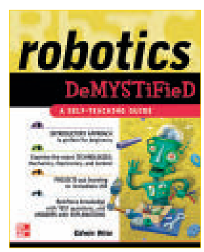
Get ready to reach into your programming toolbox and control a robot like never before! *Robot Programmer's Bonanza* is the one-stop guide for everyone from robot novices to advanced hobbyists who are ready to go beyond just building robots and start programming them to perform useful tasks.

$29.95

Robotics Demystified

by Edwin Wise

YOU DON'T NEED ARTIFICIAL INTELLIGENCE TO LEARN ROBOTICS!

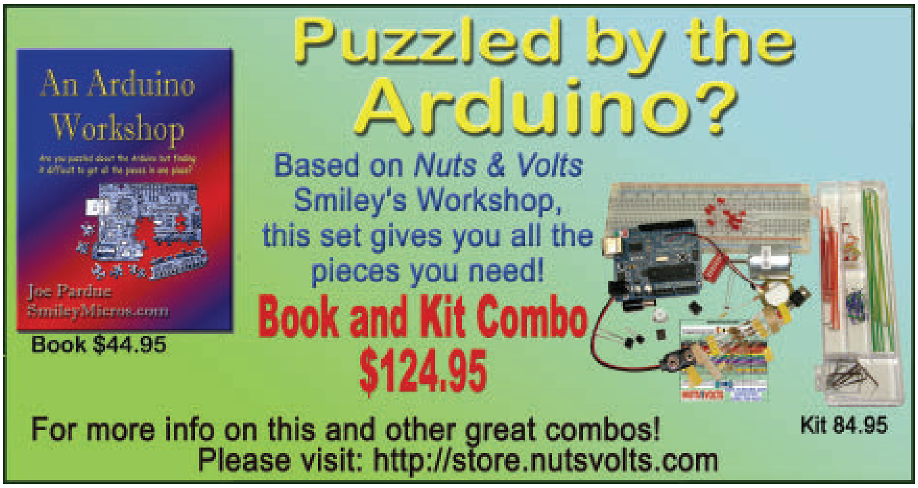


Now anyone with an interest in robotics can gain a deeper understanding — without formal training, unlimited time, or a genius IQ. In *Robotics Demystified*, expert robot builder and author Edwin Wise provides an effective and totally painless way to learn about the technologies used to build robots! **$19.95**

We accept VISA, MC, AMEX, and DISCOVER
Prices do not include shipping and may be subject to change.

Or order online
www.servomagazine.com

SPECIAL OFFERS



Forbidden LEGO

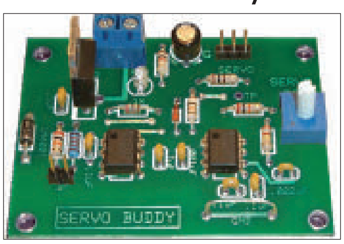
by Ulrik Pilegaard / Mike Dooley

Forbidden LEGO introduces you to the type of free-style building that LEGO's master builders do for fun in the back room. Using LEGO bricks in combination with common household materials (from rubber bands and glue to plastic spoons and ping-pong balls) along with some very unorthodox building techniques, you'll learn to create working models that LEGO would never endorse.

Reg $24.95 Sale Price $19.95

PROJECTS

The SERVO Buddy Kit



An inexpensive circuit you can build to control a servo without a microcontroller.

For more information, please check out the May 2008 issue or go to the SERVO webstore.

Includes an article reprint.

Subscriber's Price **$39.55**

Non-Subscriber's Price **$43.95**

3D LED Cube Kit

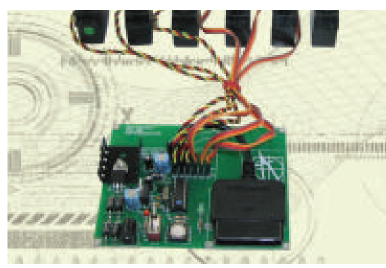
From the article "Build the 3D LED Matrix Cube" as seen in the August 2011 issue of *Nuts & Volts Magazine*.

This kit shows you how to build a really cool 3D cube with a 4 x 4 x 4 monochromatic LED matrix which has a total of 64 LEDs. The preprogrammed microcontroller that includes 29 patterns that will automatically play with a runtime of approximately 6-1/2 minutes.
Colors available: Green, Red, Yellow & Blue.
Jig and plastic cases also available.

Subscriber's Price **$57.95**

Non-Subscriber's Price **$59.95**

PS2 Servomotor Controller Kit



This kit accompanied with your own PlayStation controller will allow you to control up to six servomotors. Includes all components and instruction manual.

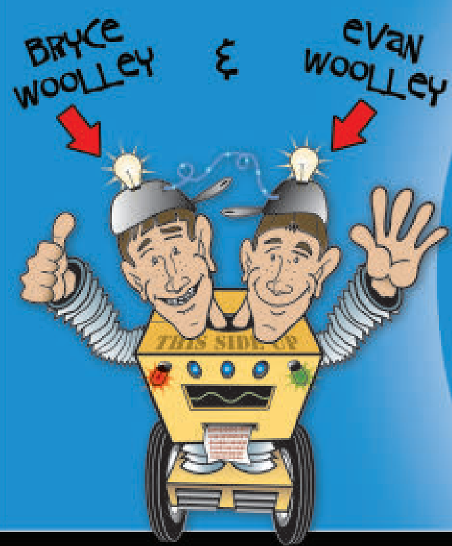
For more information, please see the February 2011 edition of *SERVO Magazine*. Assembled units available!

Subscriber's Price **$79.95**

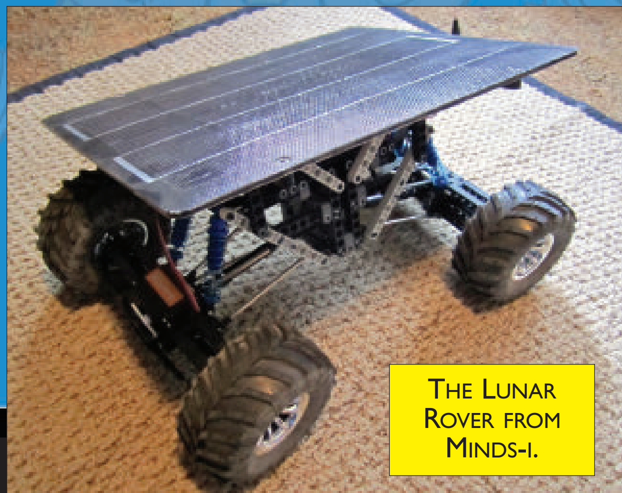
Non-Subscriber's Price **$84.95**


RIP Into Robotics This Christmas!
2 for 1 gift offer
Our special holiday offer lands YOU AND A FRIEND your very own one-year subscription to SERVO for just $24.95!
Offer expires January 31, 2013!
Lynx 6
2 Subscriptions For The Price Of One!
Send yourself and a fellow robotics nut a special gift this holiday season.
Buy or renew your own subscription to SERVO Magazine for one year and get a second subscription for your friend or loved one for FREE.
That's two (not one!) separate one-year subscriptions for a total of $24.95 (US only).
Order by phone (877) 525-2539 Or Online www.Subnow.com/SV/2For1
Use promotion code Y2FXMS

Twin Tweaks



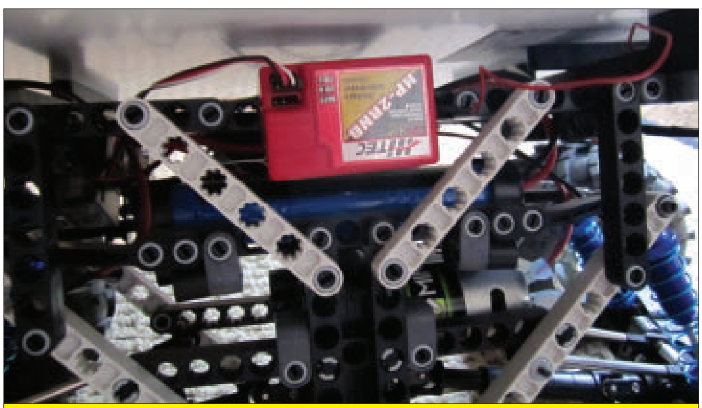
THIS MONTH: Rock and Rover



The Lunar Rover from Minds-i.

www.servomagazine.com/index.php?/magazine/article/november2012_TwinTweaks

Along with everyone else that cares about science, human progress, and things that are just plain cool, we have been awestruck and inspired by the incredible success of the Curiosity Rover and the entire Mars Space Laboratory mission thus far. We are so excited, in fact, that we determined a Curiosity-inspired project had to be the topic of our next column. Therefore, we thought this would be the perfect time to revisit the Lunar Rover 3-in-1 kit from Minds-i. We previously worked with the Lunar Rover from Minds-i in the May 2011 and June 2011 issues. The larger scale and better off-road capabilities of the Lunar Rover compared to our other kits draws a nice parallel to the larger sturdier nature of Curiosity compared to its predecessors, Spirit and Opportunity. However, our Rover was just that — a great driving base capable of handling even the most unforgiving terrain, but it lacked any additional mechanisms. While it would still be about 350 million miles away from Curiosity in sophistication, we determined that an arm mechanism would make a nice addition to the Lunar Rover.



A two-channel receiver was not going to make the cut.

Radio Heads

We're always glad for an opportunity to revisit kits, and this was the perfect opportunity to see if the Lunar Rover from Minds-i could effectively support an additional mechanism. Our Minds-i kit is the 4 x 4 Robot 3-in-1 kit — it has everything you need to make a four-wheel drive base, but for all intents and purposes it is basically a starter kit. Minds-i offers an impressive array of additional kits and sensors like the Arduino autonomous upgrade module, receiver kits, and extra structural parts. The Arduino module includes an Arduino microcontroller and a host of sensors including ultrasonic and infrared. Minds-i also offers a new transmitter set that includes a six-channel receiver and radio.

The base kit that we received has only a two-channel radio and receiver. This was fine for the base kit, which used one channel for controlling the steering servo and the other for controlling the drive motor. That left no channel for an additional mechanism, though. While we appreciated that the Minds-i folks provided another receiver and radio for order on their website, we elected to employ one of our old Futaba radios that we used with our combat robots.

The Futaba had a seven-channel receiver which would be more than enough to add a mechanism.

Quantum Reach

Now that we had more channels than we could ever want, we needed to design the mechanism. A classic device that appeals to the mechanical engineering affinities of people like us is a robotic arm. An arm would be the perfect way for our Rover to collect samples of mysterious rocks, and to discover if the Lunar Rover provided a suitable base for expansion. We thought this would also be a good exercise to test the limits of the basic kit.

As we mentioned, all we had was the 3-in-1 Rover starter kit and a structural expansion set, so we wanted to see how easily that would translate into an additional mechanism. If the basic kit and small expansion set offered plenty of material to allow a tinkerer's imagination to run wild, then we would consider the Minds-i Rover to be the perfect sort of present that would nurture a young roboticist's imagination and put them on the track to become the next JPL or NASA superstar.

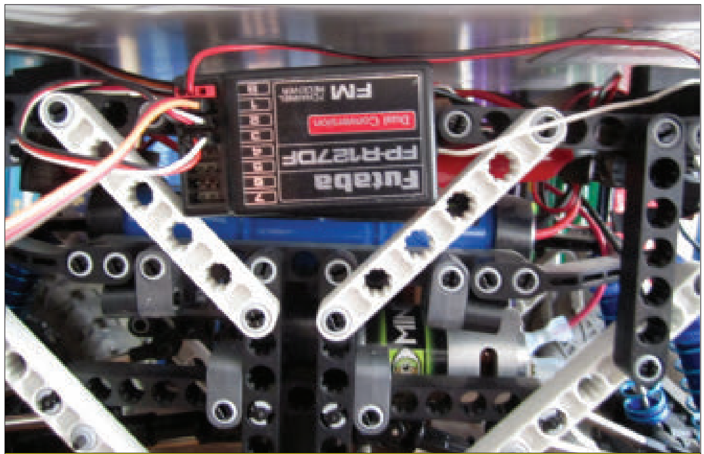
We value elegance and efficiency in our mechanical designs, and we thought it would be a creative constraint to realize our arm by using only one motor. This means fewer wires to get pinched in joints, fewer devices to draw power from the bot's single battery pack, and fewer things to program. It also means designing a mechanism that can essentially perform two motions: closing the claw and raising the arm. Thankfully, we had some design inspiration from one of our robots of yesteryear that performed a similar task.

As that design inspiration, we looked to a previous project of ours that last played a supporting role in our July 2005 column. Gog VI is a LEGO Mindstorms robot that we built for the Science Olympiad competition when we were in high school. The goal of the Robot Ramble event at the Science Olympiad was to deposit a variety of objects into a box, with the caveat that the same box determined the maximum dimensions of the robot.

We wanted to be weight conscious with the arm mechanism because the LEGO motors had limited power, and we wanted to figure out a way to articulate the claw without putting a motor on the end of a long moment arm.

The final design involved a worm gear driven by the motor that meshed with a large gear. The large gear was able to articulate both the end effector and the arm itself by directly pulling the support connected to the end effector. When the end effector opened or closed completely, it would lock against the supports for the arm, and thus the large gear would lift the entire arm.

LEGOS, of course, are renowned far and wide for their accessibility and adaptability into any design that you could possibly come up with. Would the leftovers from the Lunar Rover starter kit and a bag of extra parts be enough to build a similar mechanism?



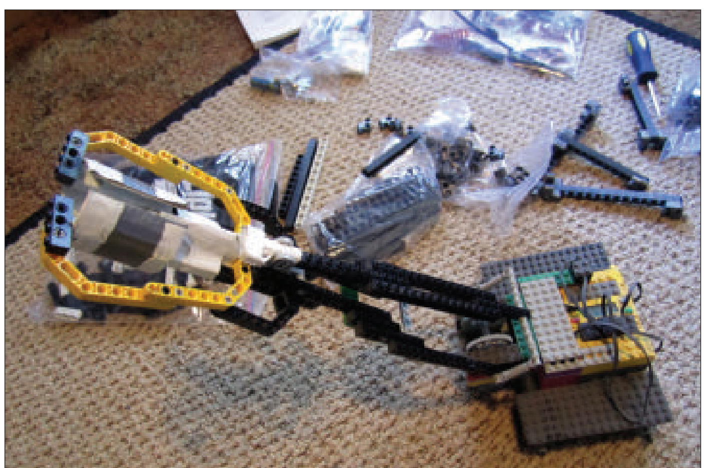
A SEVEN-CHANNEL FUTABA RECEIVER.

Mars Bars

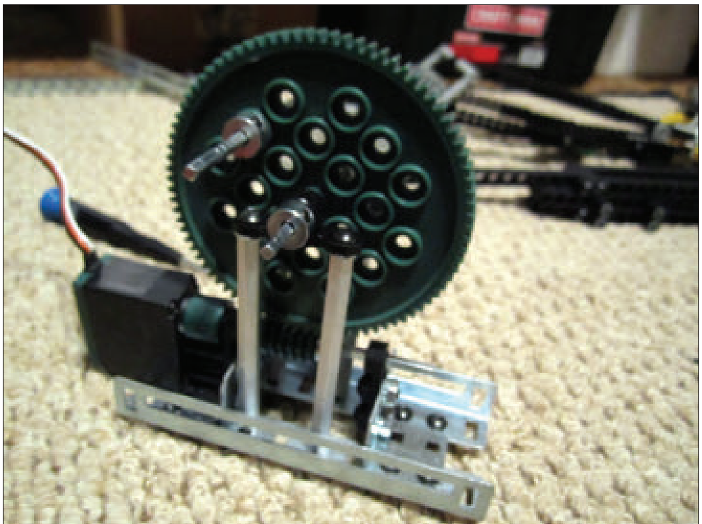
Our first instinct was to use a Savox servo to power the arm mechanism. We were concerned that scaling up the design we used on a small LEGO robot would require a ton of power – a claw on the end of a long arm is not the sort of endeavor for a servo short on torque.

By a cruel twist of fate, however, our Savox servos were AWOL from Robot Central, so we had to go to Plan B. Our next choice was to use one of the motors from the VEX Robotics design system. A VEX motor is not as torquey as the Savox servos, but it is much easier to use as the centerpiece of a mechanism – the square shafts of the VEX kit are much easier to attach to mechanisms than the sometimes fiddly holes on a plastic servo horn.

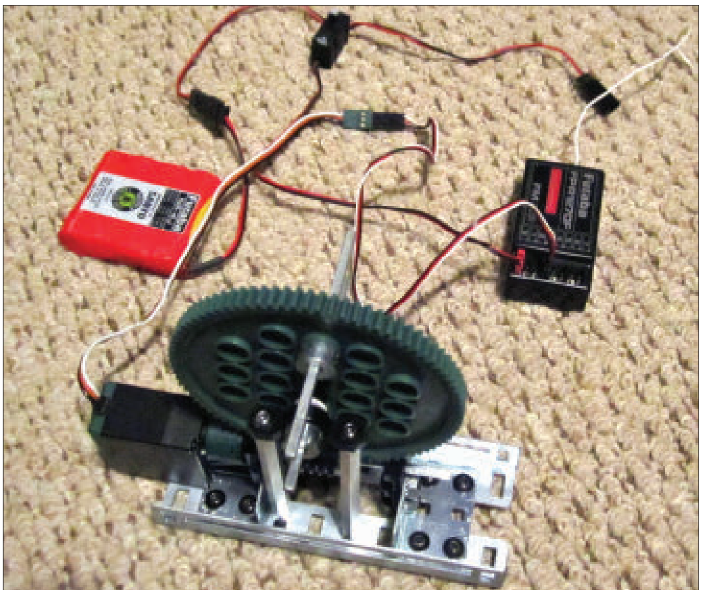
Using a VEX motor also meant it would be easy to mate to other VEX parts, namely a worm gear from the VEX gear set. Aluminum VEX frame pieces would also make for a sturdy bracket for the motor and gears. We were somewhat apprehensive about mixing kits, though. In our last project, we added the brain from a Mark III robot to a



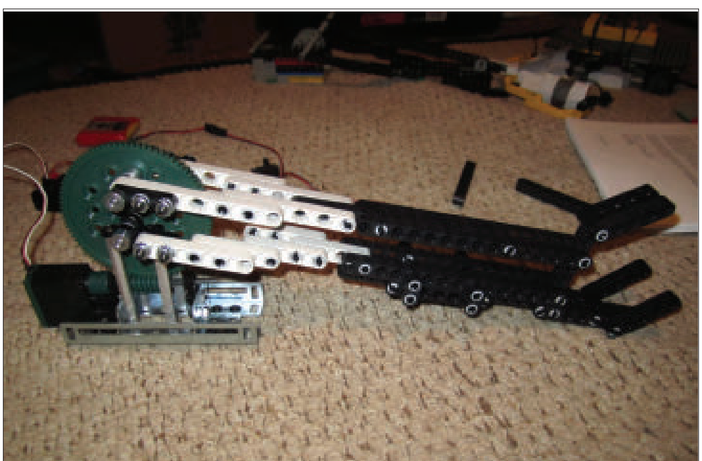
GOG VI — A DESIGN INSPIRATION.



GETTING THE WORM GEAR MECHANISM TO MESH.



INITIAL TESTING ON THE MECHANISM.



AN EARLY VERSION OF THE CLAW.

Cobra chassis from Fingertech. Despite the fact that the brain and the chassis each had four mounting places, they were not at all aligned. This only ended up being a minor headache, but the larger scale of this project could mean a larger headache.

So, it was with some trepidation that we aligned the VEX frame pieces with some of the frame pieces from the Minds-i kit. We were crestfallen to discover that the black frame pieces did not even remotely line up with the VEX pieces we wanted to couple to it. That disappointment soon turned to elation when we saw that the white frame pieces were actually perfectly aligned with the hole spacing on the VEX parts. Encouraged by this revelation, we continued to build up the arm mechanism.

After meshing the worm gear with the large gear using a VEX frame, we needed an arm and a claw for it to articulate. This is where the extra parts from the Minds-i kit grabbed the spotlight. The arm design we had in mind would include a claw with one fixed side and one moving side. The assembly that moved was essentially a kind of four bar linkage.

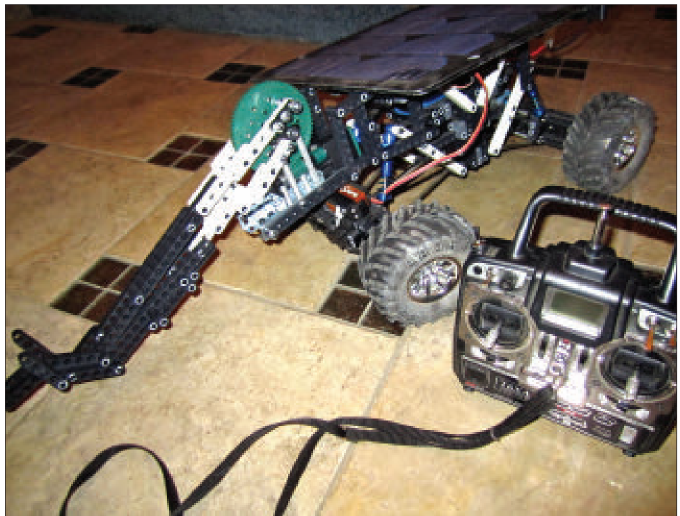
Four bar linkages are a classic device, favored for their easy design and predictable range of motion. The four bar linkage that comprised our arm was of the planar variety, because all of the joints had only one degree of freedom (i.e., they moved within a single plane). Each joint in a planar four bar linkage can be characterized as either a resolute or prismatic joint. A resolute joint is one that spins around an axis, while a prismatic joint slides along a line. All of the joints in our four bar linkage were resolute joints, making our linkage a planar quadrilateral linkage.

That's a lot of fun vocabulary, but how does it help you evaluate the effectiveness of your design, or come up with the design in the first place? Four bar linkages are so useful because they can transform one type of motion into another – rotational motion into linear motion, continuous motion into oscillation, and much more. By understanding the basics of four bar linkages, a tinkerer will never be at a loss for how to design a certain mechanism as long as you know the type of motion you want to achieve.

To illustrate: We wanted to devise a claw and arm mechanism where the claw would open and close, and where the arm would raise and lower — all with one input (one motor). The claw opening and closing is a type of non-continuous rotational motion – the moving half of the claw does not spin around completely, but rather rotates only in a limited arc. The raising and lowering of the arm is also a non-continuous rotational motion. A four bar linkage with this sort of movement is often referred to as a double rocker. A double rocker is achieved when the sum of the lengths of the shortest and longest links are greater than the sum of the lengths of the other two links. Other calculations can also be performed to determine the lengths of the arcs that the linkage will travel.

Thus, some simple addition is all the math we needed to determine the basic design of the arm and claw mechanism. The frame pieces from the Minds-i kit are

Radio upgrade.



Ready for rovering.

particularly well-suited to such a design; the multiple mounting holes on each frame piece make basic adjustments of the link dimensions quick and easy. Leftovers from the 3-in-1 Rover kit and the bag of extra parts had far more than enough frame pieces to build our arm and claw mechanism, and soon we were ready for some testing.

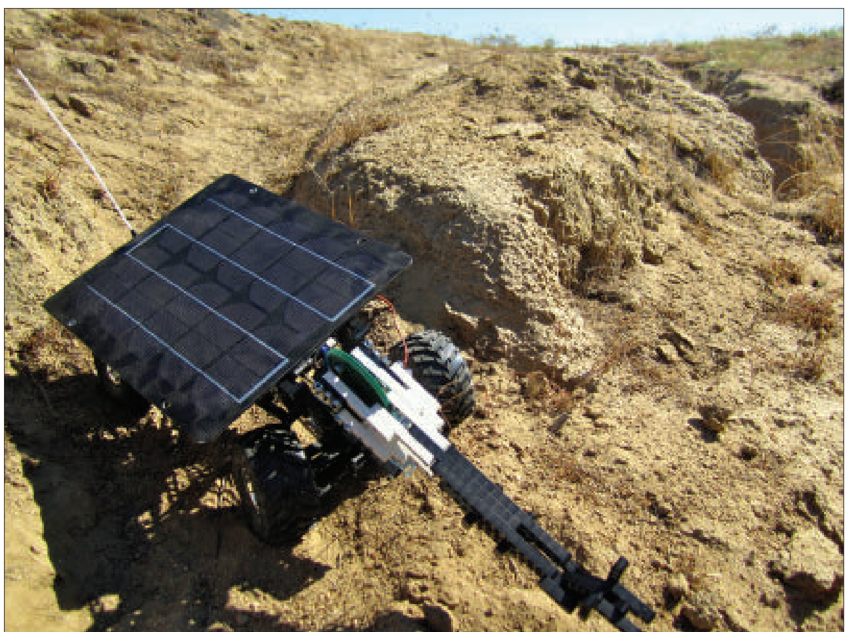
Red Planet Rover, Come On Over

Before we attached our four bar linkage to the gear mechanism, we wanted to confirm that our trusty old Futaba radio and receiver would play nicely with the VEX motor and the rest of the moving parts on the Rover itself. We wired up the mechanism – motor to receiver and battery to receiver — and fired up the Futaba. The mechanism sprang to life and the large gear began to turn ponderously, much like the relaxed pace of an ancient water wheel on a lazy stream. The slow speed was expected given the gear ratio, and we hoped that would also translate to big enough torque to raise a long arm and a rock sample at the end of it.

Now, we had a working gear train and a claw assembly. Fastening the components together was made delightfully easy by the serendipitous alignment of the Minds-i and VEX mounting holes. Before mounting the whole thing to the robot, we wanted to ensure that the claw operated the way we intended. Performing any redesigns would be much easier before attaching the arm to the bot.

When we fired up the arm, we were pleased to see that once the claw opened or closed the entire arm would raise or lower. Unfortunately, the claw opened before raising the arm, and closed before lowering it. This was the opposite of what we wanted to do, and it pretty much looked like the claw was flipping off its careless creators. When we took a second look at our mechanism, it was obvious why it was happening.

The upper part of the claw was the link in the four bar linkage, and because we only had one input, of course, the links would all move in the same direction. Thus, when the motor moved in the direction to raise the arm, the claw would open. Understanding the problem, however, also led to an understanding that it was easy to fix. Because the moving part of the claw would rotate in the same direction as the arm, we just had to make it so that "raising" the claw link corresponded to closing the claw. The fix was clear: Instead of articulating the top of the claw and leaving the bottom fixed, we should articulate the bottom of the



Navigating tough terrain with ease.


FINDING A SAMPLE.

would simply create an even larger moment arm and even more work for our hapless VEX motor.

Weighing these pros and cons, we decided to roll the dice on a front-end mount. The existing frame of the Rover provided plenty of solid mounting points, and the Minds-i frame pieces were perfect for securing the mechanism to the front of the Rover. The bulky mechanism and long arm compressed the front shocks a bit, but on the bench the Rover seemed to have no trouble holding the arm off of the ground. Whether the Rover could maintain its posture over rough terrain, however, would be the real test.

The final step before testing would be programming the radio. Programming the Futaba is really just a matter of mapping the controls, and we assigned one joystick to the control of the drive motor and the other to the control of the arm and claw. After some quick bench-testing to make sure the controls weren't reversed or otherwise disoriented, we were ready to take the bot off-road.

claw and leave the top fixed.

After a little bit of rearranging the links, we were relieved to see that the claw behaved in exactly the way we wanted it to – the claw would close before the arm would raise, and the claw would open after the arm had lowered to the ground. A working arm and claw is a great achievement, but it wouldn't be much more than an idle exercise if it remained detached from the Rover.

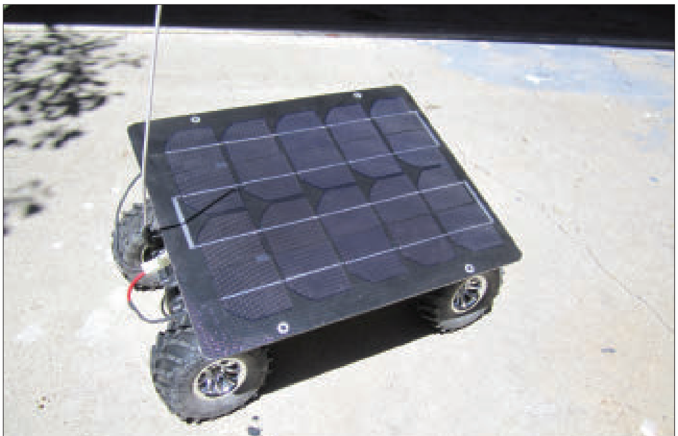
Our final task was to attach the claw to the Rover somehow. The still plentiful leftover Minds-i frame bits provided plenty of raw materials, but first we had to figure out where to put the mechanism. To properly sample alien rocks, we wanted the claw to be able to reach the ground. At its present length, that would be most easily achieved by hanging the mechanism off of the front of the robot. We were concerned that this might shift the center of gravity in ways that might have some unintended consequences.

Another option was to mount the mechanism to the top of the Rover. It wouldn't compromise the Rover's center of gravity, but this option also had drawbacks. First and foremost, mounting the mechanism on top of the Rover would require us to remove the solar panel. We really didn't like that idea, because the solar panel was not only an important source of cool factor, but also something that seemed inextricably intertwined with the bot's identity as a Rover. Mounting the mechanism to the top of the Rover would also mean that the arm in its current state could not reach the ground. We could always extend it, but that

Rover the Explorer

We were finally ready to test the Rover, but we were concerned that the bulky mechanism might compromise the agility of the bot. We were extremely impressed by the ability of the unmodified Lunar Rover to negotiate even the


TIME FOR SCIENCE!



Catching some rays to charge up.

Leaving its mark.

gnarliest off-road terrain, especially after we replaced the steering servo with a powerful Savox servo. The Rover could climb rocks, careen down slopes, and plow through obstacles as confidently as Bear Grylls after a delicious snack of exotic fauna. However, all that impressive off-roading was done without a long arm hanging off of the front of it. We were concerned that the weight of the mechanism would take a lot of the springiness out of the shocks, unbalance the Rover, and compromise the turning radius.

All of those fears, however, were quickly dissipated when we fired up the bot. It could handle rough terrain just as easily as it ever could. The only thing we had to watch out for was getting the bulky arm caught in obstacles, but the killer Savox servo provided more than enough power to allow the Rover to maneuver with ease.

With the Rover driving properly, the final test would be to see if the arm and claw could actually pick up and manipulate objects. We drove the Rover up to a suitable rock and lowered the claw. Upon hitting the ground, the claw began to open at a slow and contemplative pace because of the gear ratio.

Once the claw was open, we edged the Rover closer to our prospective sample until it looked to be within our grasp. Reversing the direction on the joystick saw the claw begin to close, and once it grasped the sample tightly, the arm slowly ascended. Once the arm was at a suitable height, we thought that a proper claw would have to pass one last test: It would need to maintain its grip on the sample as the Rover drove around. We started slowly at first, but then made full speed turns, quickly changing direction and swinging the Rover around. The sample stayed firmly in the clenched claw, and we were thrilled with the bot's success.

Final Thoughts On Our Minds

Overall, we were very impressed with the expandability of the 3-in-1 Lunar Rover kit from Minds-i. The extra parts provided plenty of raw materials for a nice four bar linkage mechanism, and the kit meshed nicely with parts from the VEX kit. The Rover was also amenable to switching out the radios.

This project was done with only parts that we happened to have on hand. With all of the expansion kits offered by Minds-i — like the Arduino autonomous upgrade kit and numerous sensors — the Lunar Rover kit is sure to fire up a roboticist's imagination and inspire them to shoot much farther than the moon. **SV**

Then and NOW

Servos

by Tom Carroll

I wrote about servos in this column back in the beginning of 2008 when the Robotis Dynamixel rotary actuators had been available for a while for the robot experimenter. I want to describe these, as well as the many other types that are for sale to robot builders. Those little plastic boxes with the rotating plastic disc or X-shaped 'horn' have made small robot design and construction easier for the beginner robot builder as well as for the experienced experimenter. Rather than describing the history of servo mechanisms from the steam age to the present, I will just concentrate on the typical servo that we might find in a hobby catalog or in advertisements here in *SERVO Magazine*.

Those of us who have built robots over the years have always known that robots do not have the strength that the public ascribes to them. Steve Austin — "The Six Million Dollar Man" — might have been able to lift tons of weight with his built-in man-made muscles, and Robbie the robot in *Forbidden Planet* (**Figure 1**) supposedly could topple a house off its foundation, but we are actually quite happy if our robots can lift a pound or two with their arms. I happily watched this TV show and movie (and many more that featured robots), knowing full well that 'real' robots did not have the power shown in the films.

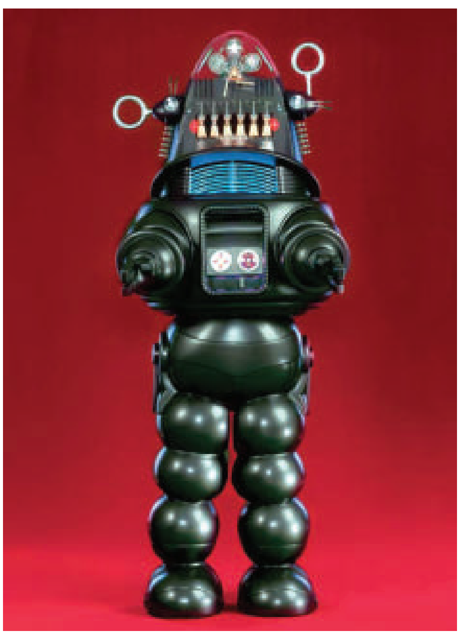


FIGURE 1. Robby the Robot.

Even human-sized industrial robots lack the raw lifting strength of a person, but make up for this inability with extreme accuracy and the capability to deliver many precision movements without stopping.

Most servos of today certainly do not have the strength that even small animals the size of our robots might have, but servos have given our smaller creations amazingly complex maneuverabilities.

Types of Servos

From just a handful of available servos three decades ago, today's servos number in the many hundreds in models, and millions are sold for all sorts of applications. There are numerous manufacturers, most of which are from overseas. I am going to review only a few manufacturer's products, such as Hitec who makes over 50 different servos — about half analog and half digital, with five of those specifically for robots. Futaba has over 130 in its line of servos, including 20 brushless and 11 in its line of S.bus programmable servos.

JR and Spektrum are among others who manufacture quality servos, but I am limited in space here, so won't get into those. I am also going to discuss some unique solutions to improving servo performance. **Figure 2** is a scene of one of my worktables showing a wide variety of servos and associated test equipment that I will detail throughout this article.

Analog vs. Digital Servos

For many years, everyone seemed quite happy with the proportional control offered by analog servos. Let's face it. These small, affordable servos were first designed for model aircraft radio control. Yes, some servos were designed around radio control boat's and car's needs, but the specifications were mostly formed around aircraft needs — especially small and light

servos. Boats could use almost any size, as could most R/C cars. Robot builders had to paw through a pile of different types in the early days to find a servo that had enough torque for their unique builds.

So, it was the analog servo that we first used. With an input pulse train of about 1.0 to 2.0 millisecond pulses, these little plastic boxes could interpret pulses from a receiver (or microcontroller) and rotate the output shaft back and forth from about 90 to 120 degrees or more, depending on the servo and its programming. With 50 20-millisecond pulse sequences per second, longer pulses would cause the output shaft to proportionally turn one way, and shorter pulses the other way. The 50 pulse sequences resulted in 50 proportional voltage pulses to the servo's motor — either one polarity or another at a variable voltage.

Internal potentiometers gave an internal feedback to the servo's circuitry as to direction and speed of the output shaft. Builders soon connected electronic speed controllers (ESCs) to the output of an R/C receiver and could control a motor with the received pulse trains. These ESCs were first used in boats and cars before being utilized by robot experimenters.

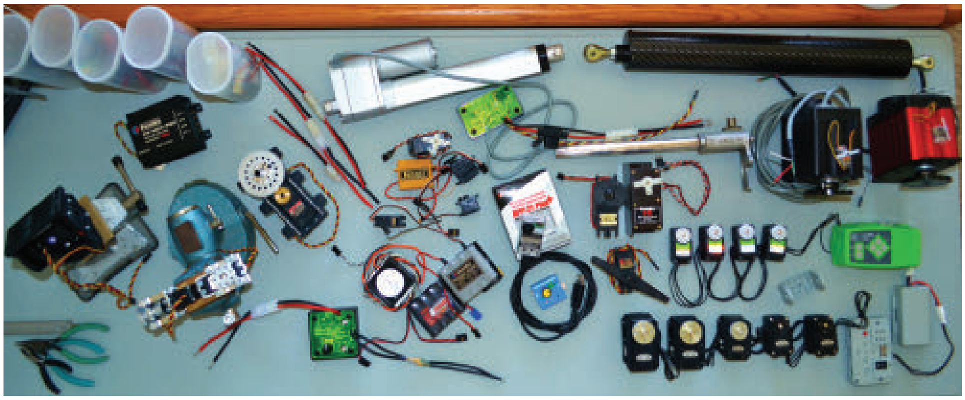


FIGURE 2. My tabletop with a wide variety of servos.

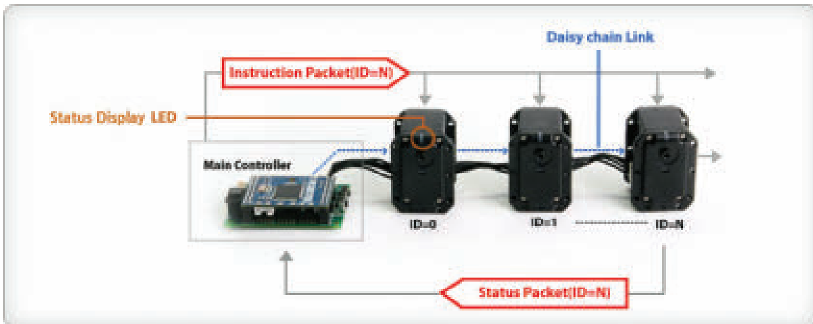


FIGURE 3. Robotis Dynamixel daisy-chain.

The Digital Servo

Digital servos look identical to their analog brethren and receive the same pulse trains, but they use a microcontroller to process the pulses into a higher frequency series of output pulses to drive the servo's motor. The resulting servo output has a more constant torque level and almost no 'dead-band' movement.

Users can feel the dead-band in an older analog servo when they try to turn the output shaft/horn with their fingers and feel a bit of slop back and forth. They can also feel the buzzing of the output caused by the lower frequency pulses, whereas the digital servo moves very little when a signal is applied; the higher frequency buzz of up to 300 Hz is negligible.

Humanoid robot designers seem to like the digital servos as they react faster and don't seem to have that Parkinson's buzz when not moving. The main disadvantage to digital servos is the higher current draw, plus they're a bit costlier.

Dynamixel and Herkulex Servos

Back in 2005, I first saw a Bioloid robot made by the Korean company, Robotis, and its use of unique servo modules known as Dynamixel rotary actuators (more on this later). Standard servos utilize a three-wire connection, a power line, a neutral power line, and a signal line. Most complex robots such as humanoid bipedal robots or hex walkers use as many as 18 or more servos, and each servo must have the two power lines and a signal line.

Builders found that they could parallel the two power lines but were still required to have 18 signal lines running from their controller — one for each servo. The Dynamixel does not require this and interconnection is daisy-chained as shown in **Figure 3**.

These servos are much more than just a regular type in that each has a built-in microcontroller with an individually addressable ID. Plus, they have the ability to send information back to the controller as to speed, torque, shaft position, temperature, voltage, and load factors. There are 13 different Dynamixels in the Robotis lineup that we'll review here.

In 2011, another Korean company — Dongbu Robot — developed a line of similar servos called Herkulex servos. Dongbu has two Herkulex servos in their line, and utilize them in a line of Hovis robots — one of which I have built. These servos use the same type of daisy-chain interconnection as the Dynamixels. I will discuss both of these servo types later.

Other Servo Factors

Other factors to take under consideration when selecting servos are the motor driving circuitry and electrical characteristics, as well as the mechanical aspects of the overall servo design. Earlier servos used what is called a three-pole motor armature with three distinct sets of wirings on three laminated metal poles.

Some servos have gone to a five-pole armature for smoother rotation and more torque. A few servo designs

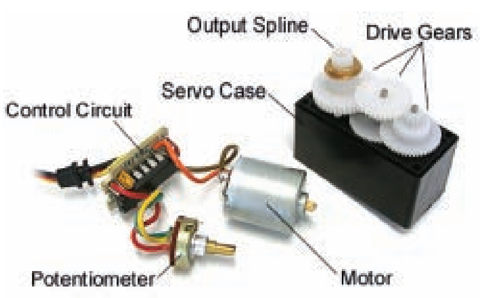


FIGURE 4. Servo breakdown (*courtesy of ServoCity*).

even use rare earth magnets for a stronger field, but this technology is usually limited to larger DC drive motors.

The newest servo motors use what is known as 'coreless' construction that dispenses with rotating the core of heavy magnets, but rotates only the wires instead. This lightweight 'basket' of wired coils accelerates and decelerates faster. Even newer motors use a brushless design that has very little internal drag from brushes, plus they last longer. These newer designs provide more torque, quicker response, and lower power requirements, but as you might expect, at a higher cost.

An important factor to some is to determine if it can be used as a drive motor for a small robot. One very popular procedure to apply to a standard servo is to retrofit it to continuous rotation. There are numerous sites on the Internet that can assist you in making a servo continuously rotate CW or CCW when a 1.5 ms to 1.0 ms signal (or a 2.0 ms to 1.5 ms signal) is applied. I won't go into the fine details as each servo is different and requires different techniques. **Figure 4** shows a typical servo opened up for modification (courtesy of ServoCity). Talk with others or search 'continuous rotating servo mod' on the Internet for help. The second consideration in servo design and selection are the mechanical aspects. Stepping up from bronze bushings to ball bearings on the output shaft allows a greater load bearing capacity. Internal metal gears allow for a greater torque output with a longer mean time between failure.

Such requirements are not important in small table-top robots such as Parallax's BoeBot, but certainly come into play when servos are used in small combat robots, load bearing arms, appendages, or sensor platforms.

Many servo manufacturers offer gear replacement sets for servos with plastic or nylon gears, so keep that in mind when deciding over the use of the more expensive metal gear servos. Output shaft seals are rarely needed for most robots unless water will be encountered in normal operations.

Servos for Robots — From Very Small to Very Large

Over the years, I have used numerous servos in many types of applications, but mostly in robotics. The servos that I am reviewing in this article are all rotary servos. I will review linear servos/actuators in another article. I have personally tested all of them for potential robot applications. Both types have applications that are best fitted for their design characteristics and I saw some features in each one that I tested that are best met by another type. Servos come in all sizes, capabilities, costs, and — most importantly — applications. A prospective robot designer should look at control signal and power requirements, as well as ways of mounting, size, torque, output shaft usability, ruggedness for their application, and affordability. If a robot builder remembers that most servos were/are designed for model aircraft uses and proceeds with their robot design with that in mind, the specific application and appropriate control methods will become apparent.

Servos come in all sizes and shapes with unique features that apply to a narrow type of robot. Hitec has a great line of robot-specific servos that have been covered in many robotics blogs, so I'll concentrate on servos other than these. I selected several Hitec and Futaba standard and digital servos, several very large Torxis servos, and several each of the Dynamixel and Herkulex actuators to test and review for this article. Just to test torque, I used a very long armed horn attached to the servo that had a 1.7 inch arm length that could represent a 3.4 inch diameter reel. Very crude, but effective. So, for example, if the servo could lift 10 ounces at the 1.7" length, that represented 17 oz-in of torque. I used several other jury-rigged arms of 3", 5", and 10" in length with some brass balance weights from a set that I had as a reference mass.

Since my weights were in grams from one to 500, I had to do a bit of calculating (28.35 G = 1.0 oz/.03527 oz = 1.0 G) to result in oz-in which is what most US experimenters use. I used the two different servo testers shown in **Figures 5A** and **5B** as the servo drivers and programmers, as well as a transmitter/receiver setup.

(I used the Hitec HPP-21 Plus programmable servo tester and an inexpensive $3 import.)

Tiny Servos

Some very popular servos for tiny robots weigh only a few grams, such as the Hitec HS-5035HD digital ultra nano servo shown in **Figure 6** that is perfect for small robot claws/end-effectors.

FIGURE 5A. Hitec HPP 21-plus digital servo programmer.

FIGURE 5B. Inexpensive imported servo tester.

Tom Carroll can be reached at TWCarroll@aol.com.

They can even be modified as continuously rotating for very small robots. Weighing in at 4.5 grams, with 11 oz-in of torque, this 4.8V servo is tiny and mighty, and two of them could fit end-to-end in 1-1/8" to function as a little robot's drive motors. I plan to use these in a very tiny robot claw.

Futaba also has two mighty-mites: the S3153MG (metal gear) shown in **Figure 7**, and the S3154 that has a bit over 3/4" body depth each, and over 20 oz-in of torque. They weigh in at 0.4 oz and 0.27 oz, respectively. These two servos and the Hitec nano also make excellent sensor movers for small claw/arm servos. I want to complete a small four-axis arm/claw using these four servos to demo at some of our robotics meetings soon.

FIGURE 6. Hitec HS-5035HD three gram ultra nano servo.

FIGURE 7. Futaba 3153MG micro metal gear servo.

Small Servos

The Futaba S3150 shown in **Figure 8** is a small, 0.81 oz servo with an impressive 51.4 oz-in of torque. I was testing it as a possible small robot arm/claw motor. At 1.14" x 1.18" x 0.43" thick, it is easily contained in a small arm. The Hitec HS-7115TH shown in **Figure 9** is ultra slim at only 0.32" thick, and has over 54 oz-in of torque at 7.4 volts.

Another excellent slim servo is the Hitec HS-5125MG that is considered a 'wing' servo as it is only 0.39" thick. It's great to also hide within a small robot's arm structure. It has over 41 oz-in of torque and is programmable.

Middle-Range Servos

Many servos fall into the category of what I am calling 'mid-range.' The Hitec HS-5565MH shown in **Figure 10** is a standard servo that can operate at 6-7.4 volts with a torque of up to 194.4 oz-in; it weighs in at only a bit over two ounces. It is programmable and can operate off two Li-Po cells. The Hitec HS-645MG shown in **Figure 11** is another great mid-range servo with a 1.94 ounce (55 grams) weight and 133 oz-in of torque in a 1.6" x .77" x 1.5" plastic case.

These are great robot servos (and are the two mounted in the vise in the **Figure 2** photo). They have ServoCity servo blocks mounted on them. These mounts offer several advantages over a basic servo in that they provide a much more sturdy mounting capacity than attachment to a plastic case.

A much more important feature is the servo block's ability to take 90% of the strain off the servo's shaft and bearings. The outboard aluminum bracket just under the flanged output horn contains a large bearing. **Figure 12** shows a close-up. This allows the small servo to have a large reel, rotating arm, or arm/leg joint to be directly attached. It would be hard to machine these in a shop for a cost lower than ServoCity's price.

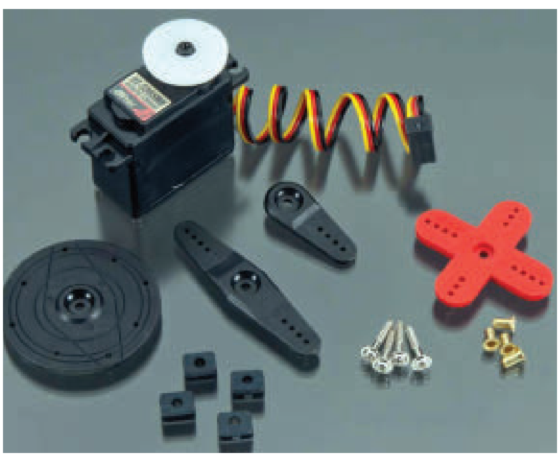


FIGURE 10. Hitec HS-5565HM with horn assortment.

The Futaba S9352HV shown in **Figure 13** is a 6-7.4 'high voltage' servo that has an impressive 305.6 oz-in of torque at 7.4 volts, though I tested it at 7.2 volts and it seemed to have the same torque capacity.

Figure 14 shows a Futaba S9156 high torque (340.3 oz-in) servo that has metal gears. Operating at 6.0 volts (NiCd battery use only) and weighing 2.22 oz, this is one powerful coreless motor servo. It is well adapted to shoulder robot arm actuation where a 17 inch arm length could lift 20 oz if the arm's weight is spring compensated. It actually bent a metal

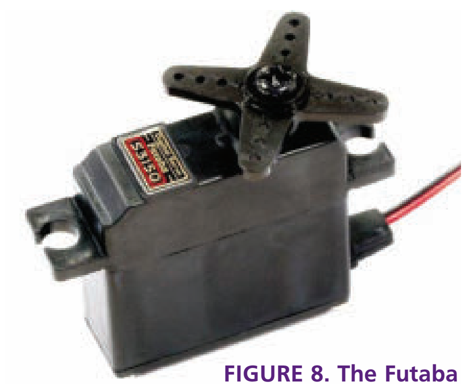


FIGURE 8. The Futaba S3150 is a powerful mid-range servo.

FIGURE 9. Hitec HS-7115TH.

FIGURE 11. Hitec HS-645MG high torque servo.

FIGURE 12. Servo block from ServoCity.

test arm rod that I was using to measure the torque with the lab balance weights. The Hitec HS-M7990TH monster torque servo shown in **Figure 15** weighs only 2.78 oz with an aluminum heatsink case, but has a massive 500 oz-in of torque at 6.0 volts. With titanium gears and dual ball bearings, a magnetic rotary encoder, and programming capability, this servo is perfect for large robots — either for leg or upper arm actuation. It exceeded my torque measurement capability and did not even slow down when grasped by my fingers.

Very Large Servos

ServoCity makes what is called a servo power gearbox shown in **Figure 16**. This particular setup used a Hitec HS-785HB sailboat winch servo with 3-1/2 revolution capability to drive up to a 7:1 gear ratio to obtain 1,281 oz-in of torque and a full 180º of rotation. The model that I have tested has a 3:1 ratio and a 420º rotation for 549 oz-in of torque. For $120, you cannot find a servo anywhere with these capabilities, nor machine the brackets with the large ball bearings to take large loads. ServoCity (through its parent company, RobotZone) has a large variety of gear and mounting accessories for many servos.

Invenscience makes the Torxis i00800 shown in **Figure 17**, and the i01855 (industrial version) super high torque R/C servos with an ultimate torque of 3,200 oz-in. As with most very large servos, this one requires a separate 12 VDC supply at three amps capacity, but uses a standard R/C PWM input. It can travel 90º with an R/C input or up to 270º with a microcontroller's input.

Though a bit slower than a small servo, it has a 60º travel in .5 seconds. It is all metal gear construction with dual roller bearings on the output shaft and an aluminum 6061-T6 horn. These are very massive and powerful servos that can handle any human-sized robot's requirements. In testing them, I had to be very careful of any pinching situation at the output horn.

Rotary Actuator Servos from Robotis

The Dynamixel rotary actuator series that I mentioned earlier has become one of the most popular servos for advanced robotics experimentation. The basic AX-12 actuator of the group is at the lower right in the two rows of actuators shown in **Figure 18**. It has a holding torque of 229 oz-in and operates from 7.2 to 12 VDC. Operating on a TTL half duplex async serial protocol through the daisy-chain interconnection, this intelligent actuator is the servo for all 18 DOF in the Bioloid Premium kit.

Figure 19 shows the AX-12A and 10 of the different actuators in a speed vs. torque graph. The Robotis website, as well as the CrustCrawler and Trossen Robotics websites, have extensive information on these amazing servos.

The AX-12A and all of the AX and more advanced MX series can be mixed as all can operate on a single 12 VDC supply. The servo's built-in microcontroller handles sensor management and position control. Each actuator can track its speed, temperature, shaft position, voltage, and load. The most advanced of the series (shown on the bottom left) is the MX-106T with 1,416 oz-in of torque and an onboard 32-bit, 72 MHz Cortex M3 controller with a PID control algorithm (Proportional Integral Derivative or 'three-term control'), and a contact-less magnetic encoder that can operate at up to 3 MB/s with the new TTL 2.0 bus structure.

I am very impressed with this whole series which I have been experimenting with using the Bioloid's controller and various shop-made add-ons to test torque and speed. I have found the AX-12A actuators to be faultless in operation with the Bioloid Premium kit (which I will review very soon). The Dynamixel AX-12A and AX-18 are plastic geared actuators with a cored motor.

The Dynamixel MX-28, MX-64T, and MX-106T are

FIGURE 13. Futaba 9352HV very high torque servo.

FIGURE 14. Futaba S9156 even higher torque servo.

FIGURE 15. Hitec HS-7990TH 500 oz-in torque servo.

FIGURE 16. ServoCity SPG785 power gearbox.

metal geared with a Maxon RE-max motor. RE-max is Maxon's economical rare-earth magnet motor in their RE line (not the real estate office chain). The TTL and RS-485 serial communications protocol allows for a daisy-chain bus structure that is ideal for humanoid and hex/quad robots that require many servos with different speed, motion, and torque settings. Universities and robotics research labs around the world have adopted the Robotis line of Dynamixel rotary actuators. Go to the aforementioned sites for much more detailed information.

Herkulex Servos from Dongbu Robot

The Herkulex servos that were introduced last year have similar features, operating capabilities, and sizes to the Dynamixel actuators. They are the green servos shown in the top row of **Figure 18** (three DRS-0101s and one DRS-0201 servo). The Herkulex servo does have a 'multi-drop' TTL full duplex UART serial communications protocol using the same type of daisy-chain interconnections. There are two servos in the Herkulex series: the DRS-0101 and the DRS-0201. The 0101 weighs 45 grams, and both are a bit more compact than the AX-12A — the smallest of the Dynamixels. Whereas the Dynamixel AX and MX series have external mounting flanges around the edge of the actuators, both of the Herkulex servos are identical in size and smaller than the Dynamixels. They do not use external flanges for mounting, but rather have tapped holes for screws.

Some users have said that they like the holes-only feature after using the Dynamixels. I think that the smaller size of the Herkulex will lend itself to some smaller robot arm designs. I've tested them and they do have a lot of torque; the DRS-0101 has almost 169 oz-in and a cored motor as stated in the literature. The DRS-0201 has almost 334 oz-in of torque and a coreless motor — both at 7.4 volts. Operating from a two cell LiPo 7.2 volt battery should be only a little less. I would certainly say that advanced robot experimenters could make use of either company's products, depending on the particular application. I am impressed with the Herkulex servos and in a few years, their support and distribution most likely will be as good as with Robotis. The Hovis robots and Herkulex servos are certainly a great start.

FIGURE 17. Torxis i00800 super high torque servo.

Final Thoughts

Servos for robots are a critical component of many designs. I have run several types of tests on the servos — many were a quick jury-rigged setup — so I advise potential customers to go to manufacturer's sites, speak with reps, and — most importantly — talk with those who have used more than the basic standard servo in robot applications. Don't overlook Hitec's line of robot servos, either.

I have spent many hours talking with folks from the many companies represented here. I would like to thank Lauren Lewis of ServoCity, Suzanne Lepine of Hitec, Hiram Crisler and Carol Pesch of Futaba/Hobbico, Collin Lewis of Invenscience, Daniel Hwang of Dongbu Robot, Jinwook Kim of Robotis, and many others who assisted me in this endeavor. I will see you next time with an article about linear servos/actuators. **SV**

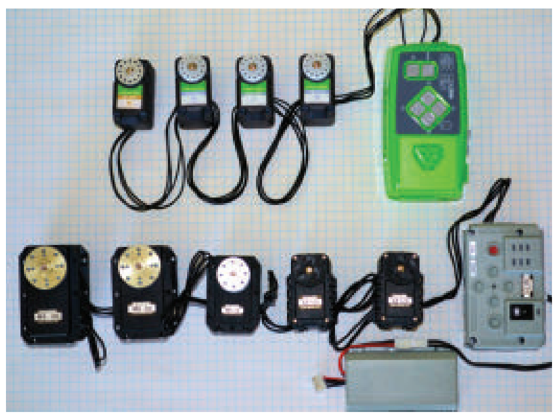


FIGURE 18. Herkulex and Dynamixel actuators.

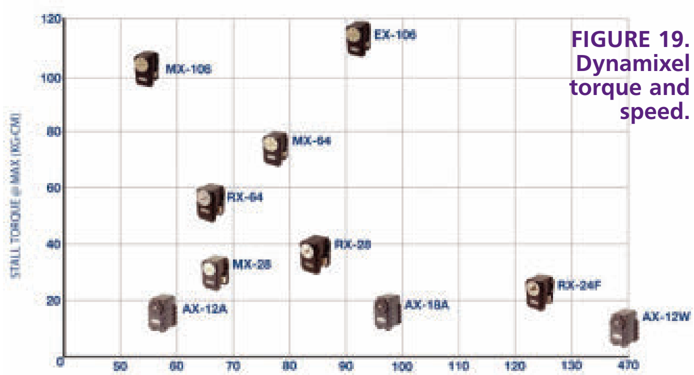
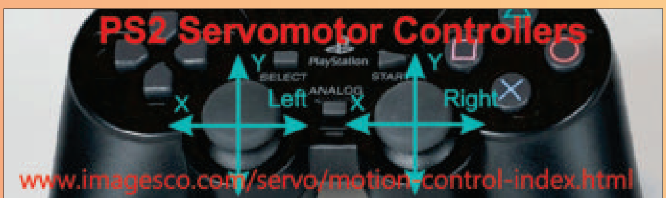
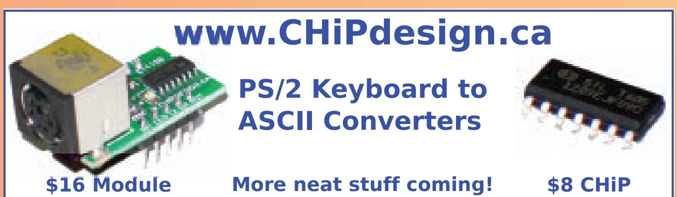


FIGURE 19. Dynamixel torque and speed.

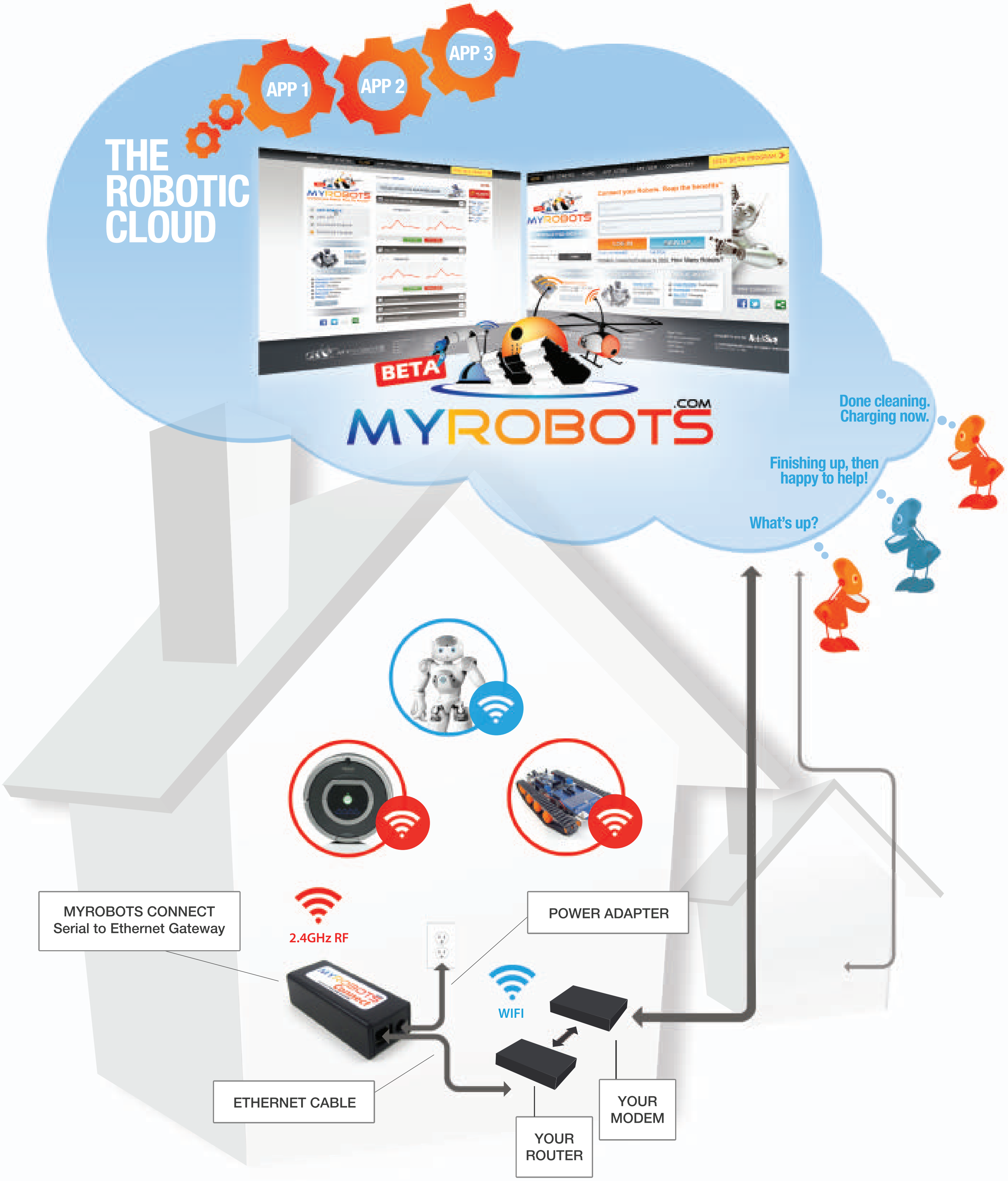
ROBO-LINKS





For the finest in robots, parts, and services, go to www.servomagazine.com and click on Robo-Links.

Advertiser Index



Connect your robots. Reap the benefits ™

MyRobots.com strives to make cloud robotics a reality accessible to everyone and everything by enabling all robots and smart devices to connect to the Internet.

You can think of MyRobots.com as a social network for robots and smart devices. In the same way humans benefit from socializing, collaborating and sharing, robots can benefit from interacting and sharing their sensors' information, which provides insight on their current state, and allows them to be controlled and monitored remotely.

Through the **MyRobots App Store**, robots augment their capabilities so they can transcend their limitations, allowing them to do more than what they were originally designed for. This means robot owners can reap the full benefits of their robots.

Brought to you by RobotShop.com Robotics at Your Service!®

Powered by ThingSpeak

DYNAMIXEL PRO

The Solution for Full Scaled Robots

High Power, High Precision

Up to 200W BLDC motor and up to 500,000
pulse per turn (4096 absolute resolution)

Full Modular Solution

Ease of implementation (no additional design for parts needed)
Ease of maintenance (simple part swapping)

Sophisticated Control Algorithms

Position and speed input commands
with dual-loop current control

Novel Gear Reduction System

High torque output, light weight,
with high impact resistance

Data		H42-20-S300-R		H54-100-S500-R		H54-200-S500-R	
		Unit	Data	Unit	Data	Unit	Data
Rated voltage		V	24	V	24	V	24
No load speed		RPM	28.3	RPM	35.2	RPM	35
No load current		A	0.61	A	1.06	A	1.18
Continuous operation	Speed	RPM	15.59	RPM	32.7	RPM	32.1
	Torque	Nm	5.596	Nm	21.142	Nm	39.131
Maximum output power		W	23.64	W	144.58	W	262.66
Resolution		Step/turn	304,000	Step/turn	502,000	Step/turn	502,000
Gear ratio		-	304	-	502	-	502
Backlash		arcmin	3.5	arcmin	3.5	arcmin	3.8
Interface		-	RS-485 / CAN	-	RS-485 / CAN	-	RS-485 / CAN
Operating temperature		˚C	5~55	˚C	5~55	˚C	5~55

DARwIn-OP

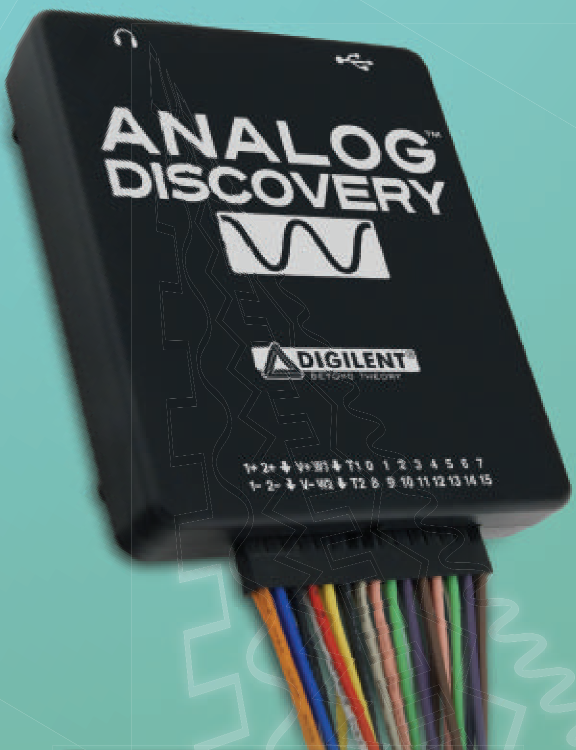
ROBOTIS
www.robotis.com

USA E-mail : america@robotis.com, Tel : +1-949-333-3635
JAPAN E-mail : japan@robotis.com, Tel : +81-3-4330-3660
Etc. E-mail : contactus2@robotis.com, Tel : +82-70-8671-2609


SERVOCITY™
We have the parts for your ideas!
hubs & adaptors
new set screw hubs
new clamping hubs
tube clamp hubs
new 90° tube clamp hubs
1" bore clamp hubs
lightweight set screw hubs
new hub spacers
clamping collars
bearings & bushings
ultra precision ball bearings
thrust bearings
ball bearing turntables
new steel shims
miniature ball bearings (thin wall)
precision bronze bushings
precision stainless steel shafting
hollow stainless steel tubing
1/8" to 1/2" available
hollow aluminum tubing
shafting & tubing
new precision stainless steel D-shafting
clamping shaft couplers
linkages & connectors
plastic ball linkages
threaded pushrods
threaded rods
metal clevis links
dogbone universal couplers
universal joints
new aluminum hub gears
acetal hub sprockets
brass pinion gears
carbon pinion gears
gears & sprockets
servo mount sprockets
metal pinion gears
servo mount gears
acetal hub gears
new
plain bore sprockets
new items!
new quarter scale SERVOblocks BY ROBOTZONE
Increase the load-bearing capabilities of your servo!
pulleys & belts
timing hub pulleys
timing pinion pulleys
timing belts
smooth belts
new smooth hub pulleys
Want to see our entire selection of mechanical parts? Visit us online!
www.servocity.com
and much more!
aluminum clamping motor mount
new 7^{99}
vertical shaft worm-drive gearbox
new 59^{99}
stacker robot kit
great beginner bot!
new 139^{99}
digital manual speed controller
new 79^{99}
mega servo
(4,000 oz-in)
new 529^{99}
SPT400 tilt system
starting at
new 94^{98}
micro gearmotors
(9 speeds available)
new 14^{99}
SAVE!
10^{00} OFF
Your purchase of $50 or more
DISCOUNT CODE: SMNOV-5
Limit one coupon per order
EXPIRES 12-31-12
You Tube
www.ServoCity.com

ANALOG™ DISCOVERY

Portable Analog Circuit Design Kit



Turn your PC into a powerful circuit learning platform that can measure, visualize, analyze, record and control mixed signal circuits! The Analog Discovery is small enough to fit in your pocket and costs less than a textbook, but it is powerful enough to replace a stack of lab equipment. Powered by a Hi-Speed USB port and the free WaveForms software, the Analog Discovery lets you build and test analog and digital circuits outside of the lab.

Designed in cooperation with:

ANALOG DEVICES

XILINX®

Dual Channel Oscilloscope

- Fully differential inputs, up to 100 MSPS, 14-bit ADC, up to 16KSa/channel memory
- 1MΩ/24pF inputs, ±20V Max
- Real-time FFTs, X-Y plots, complex math and measurements on all channels
- 5 MHz analog signal bandwidth

Dual channel Function Generator

- Single-ended outputs, up to 100 MSPS, 14-bit D/A, up to 16KSa/channel memory
- Arbitrary and pre-defined waves up to ±5V
- Sweeps, envelopes, AM and FM modulation
- Bode plot feature using standard, Nyquist & Nichols coordinates
- Can drive 50Ω

Digital I/O

- 16 signals, 100MSPS, 4KSa/channel memory
- Any signal can be input (logic analyzer) or output (pattern generator or virtual I/O)
- Supports cross-triggers with scope channels

Power Supplies

- Two fixed ±5V, 50mA

Powered by WaveForms™

$99.00 (US student price)

$159.00 (academic price)

$199.00 (List price)

For courseware & more information, please visit:
www.digilentinc.com/AnalogDiscovery

DIGILENT®
BEYOND THEORY